

Jacques Blanc-Talon Don Bone
Wilfried Philips Dan Popescu
Paul Scheunders (Eds.)

LNCS 6475

Advanced Concepts for Intelligent Vision Systems

12th International Conference, ACIVS 2010
Sydney, Australia, December 2010
Proceedings, Part II

2
Part II

 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

Alfred Kobsa

University of California, Irvine, CA, 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

TU Dortmund University, Germany

Madhu Sudan

Microsoft Research, Cambridge, MA, USA

Demetri Terzopoulos

University of California, Los Angeles, CA, USA

Doug Tygar

University of California, Berkeley, CA, USA

Gerhard Weikum

Max Planck Institute for Informatics, Saarbruecken, Germany

Jacques Blanc-Talon Don Bone
Wilfried Philips Dan Popescu
Paul Scheunders (Eds.)

Advanced Concepts for Intelligent Vision Systems

12th International Conference, ACIVS 2010
Sydney, Australia, December 13-16, 2010
Proceedings, Part II

Volume Editors

Jacques Blanc-Talon
DGA/D4S/MRIS
94114 Arcueil, France
E-mail: jacques.blanc-talon@dga.defense.gouv.fr

Don Bone
Canon Information Systems Research Australia
Sydney, NSW 2113, Australia
E-mail: don.bone@cisra.canon.com.au

Wilfried Philips
Ghent University
B9000 Ghent, Belgium
E-mail: philips@telin.UGent.be

Dan Popescu
CSIRO ICT Centre
Epping, NSW 1710, Sydney, Australia
E-mail: dan.popescu@csiro.au

Paul Scheunders
University of Antwerp
2610 Wilrijk, Belgium
E-mail: Paul.Scheunders@ua.ac.be

Library of Congress Control Number: 2010940504

CR Subject Classification (1998): I.4, I.5, C.2, I.2, I.2.10, H.4

LNCS Sublibrary: SL 6 – Image Processing, Computer Vision, Pattern Recognition, and Graphics

ISSN 0302-9743
ISBN-10 3-642-17690-9 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-17690-6 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.com

© Springer-Verlag Berlin Heidelberg 2010
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180

Preface

This volume collects the papers accepted for presentation at the 12th International Conference on “Advanced Concepts for Intelligent Vision Systems” (ACIVS 2010). Following the first meeting in Baden-Baden (Germany) in 1999, which was part of a large multiconference, the ACIVS conference then developed into an independent scientific event and has ever since maintained the tradition of being a single track conference. ACIVS 2010 attracted computer scientists from 29 different countries, mostly from Europe, Australia, and the USA, but also from Asia.

Although ACIVS is a conference on all areas of image and video processing, submissions tend to gather within certain major fields of interest. This year 3D and depth processing and computer vision and surveillance were popular topics. Noteworthy are the growing number of papers related to theoretical developments. We would like to thank the invited speakers Mubarak Shah (University of Central Florida), Richard Kleihorst (VITO, Belgium), Richard Hartley (Australian National University), and David Suter (Adelaide University) for their valuable contributions.

A conference like ACIVS would not be feasible without the concerted effort of many people and support of various institutions. The paper submission and review procedure was carried out electronically and a minimum of two reviewers were assigned to each paper. From 144 submissions, 39 were selected for oral presentation and 39 as posters. A large and energetic Program Committee, helped by additional referees (111 people in total) – listed on the following pages – completed the long and demanding review process. We would like to thank all of them for their timely and high-quality reviews. Also, we would like to thank our sponsors, CSIRO, Ghent University, CiSRA, NICTA, Antwerp University, Philips Research, Barco, and DSP-Valley for their valuable support.

Last but not least, we would like to thank all the participants who trusted in our ability to organize this conference for the 12th time. We hope they attended a stimulating scientific event and enjoyed the atmosphere of the ACIVS social events in the city of Sydney.

September 2010

J. Blanc-Talon
D. Bone
D. Popescu
W. Philips
P. Scheunders

Organization

ACIVS 2010 was organized by CSIRO and Ghent University.

Steering Committee

Jacques Blanc-Talon	DGA, France
Wilfried Philips	Ghent University - IBBT, Belgium
Dan Popescu	CSIRO, Australia
Paul Scheunders	University of Antwerp, Belgium

Organizing Committee

Don Bone	Canon Information Systems Research Australia, Australia
Russell Connally	Macquarie University, Australia
Dan Popescu	CSIRO, Australia

Sponsors

ACIVS 2010 was sponsored by the following organizations:

- CSIRO
- Ghent University
- CiSRA
- NICTA
- Philips Research
- Barco
- DSP Valley
- Antwerp University

Program Committee

Hamid Aghajan	Stanford University, USA
Marc Antonini	Université de Nice Sophia Antipolis, France
Laure Blanc-Feraud	INRIA, France
Philippe Bolon	University of Savoie, France
Salah Bourennane	Ecole Centrale de Marseille, France
Dumitru Burdescu	University of Craiova, Romania
Umberto Castellani	Università degli Studi di Verona, Italy
Jocelyn Chanussot	INPG, France
Pamela Cosman	University of California at San Diego, USA
Yves D'Asseler	Ghent University, Belgium
Jennifer Davidson	Iowa State University, USA
Arturo de la Escalera Hueso	Universidad Carlos III de Madrid, Spain
Touradj Ebrahimi	Ecole Polytechnique Fédérale de Lausanne, Switzerland
Christine Fernandez-Maloigne	Université de Poitiers, France
Don Fraser	Australian Defence Force Academy, Australia
Jerome Gilles	UCLA, USA
Georgy Gimel'farb	The University of Auckland, New Zealand
Markku Hauta-Kasari	University of Eastern Finland, Finland
Mark Hedley	CSIRO ICT Centre, Australia
Dimitris Iakovidis	University of Athens, Greece
Tianzi Jiang	The Chinese Academy of Sciences, China
Arto Kaarna	Lappeenranta University of Technology, Finland
Andrzej Kasinski	Poznan University of Technology, Poland
Richard Kleihorst	VITO, Belgium
Nikos Komodakis	University of Crete, Crete
Murat Kunt	EPFL, Switzerland
Hideo Kuroda	FPT University, Vietnam
Olivier Laligant	IUT Le Creusot, France
Kenneth Lam	The Hong Kong Polytechnic University, China
Peter Lambert	Ghent University, Belgium
Alessandro Ledda	Artesis University College, Belgium
Maylor Leung	Nanyang Technological University, Singapore
Yue Li	CSIRO ICT Centre, Australia
Brian Lovell	University of Queensland, Australia
Guojun Lu	Monash University, Australia
Anthony Maeder	University of Western Sydney, Australia
Xavier Maldague	Université de Laval, Canada
Joseph Mariani	Université Paris VI, Paris XI, France
Gérard Medioni	USC/IRIS, USA

Fabrice Mériaudeau	IUT Le Creusot, France
Alfred Mertins	Universität zu Lübeck, Germany
Jean Meunier	Université de Montréal, Canada
Amar Mitiche	INRS, Canada
Rafael Molina	Universidad de Granada, Spain
Adrian Munteanu	Vrije Universiteit Brussel, Belgium
Frank Nielsen	Ecole Polytechnique - Sony CSL, France
Fernando Pereira	Instituto Superior Técnico, Portugal
Stuart Perry	Canon Information Systems Research Australia, Australia
Massimo Piccardi	University of Technology Sydney, Australia
Aleksandra Pizurica	Ghent University - IBBT, Belgium
William Puech	LIRMM, France
Gianni Ramponi	Trieste University, Italy
Paolo Remagnino	Kingston University, UK
Luis Salgado Alvarez de Sotomayor	Universidad Politécnica de Madrid, Spain
Guna Seetharaman	AFRL, USA
Andrzej Sluzek	Nanyang Technological University, Singapore
Changming Sun	CSIRO, CMIS, Australia
Hugues Talbot	ESIEE, France
Frederic Truchetet	Université de Bourgogne, France
Marc Van Droogenbroeck	University of Liège, Belgium
Peter Veelaert	University College Ghent, Belgium
Gerald Zauner	Fachhochschule Oberösterreich, Austria
Pavel Zembek	Brno University of Technology, Czech Republic
Djemel Ziou	Sherbrooke University, Canada

Reviewers

Hamid Aghajan	Stanford University, USA
Marc Antonini	Université de Nice Sophia Antipolis, France
Sileye Ba	Telecom Bretagne, France
Etienne Baudrier	University of Strasbourg, France
Rik Bellens	Ghent University, Belgium
Jacques Blanc-Talon	DGA, France
Philippe Bolon	University of Savoie, France
Don Bone	Canon Information Systems Research Australia, Australia
Patrick Bonnin	Université de Versailles Saint Quentin, France
Alberto Borghese	University of Milan, Italy
Salah Bourennane	Ecole Centrale de Marseille, France
Dumitru Burdescu	University of Craiova, Romania
Alice Caplier	Université de Grenoble, France

Umberto Castellani	Università degli Studi di Verona, Italy
Frédéric Champagnat	ONERA, France
Jocelyn Chanussot	INPG, France
François Chaumette	IRISA, France
Wojtek Chojnacki	Adelaide University, Australia
Pamela Cosman	University of California at San Diego, USA
Yves D'Asseler	Ghent University, Belgium
Matthew Dailey	Asian Institute of Technology, Thailand
Fofi David	University of Bourgogne, France
Jennifer Davidson	Iowa State University, USA
Jan De Cock	Ghent University, Belgium
Arturo de la Escalera Hueso	Universidad Carlos III de Madrid, Spain
Jonas De Vylder	Ghent University, Belgium
Luigi Di Stefano	University of Bologna, Italy
Koen Douterloigne	Ghent University, Belgium
Marc Ebner	Eberhard Karls Universität Tübingen, Germany
Hazim Ekenel	Karlsruhe Institute of Technology, Germany
Christine Fernandez-Maloigne	Université de Poitiers, France
Yohan Fougerolle	University of Bourgogne, France
Don Fraser	Australian Defence Force Academy, Australia
Jerome Gilles	UCLA, USA
Georgy Gimel'farb	The University of Auckland, New Zealand
Aurélien Godin	DGA, France
Werner Goeman	Ghent University, Belgium
Bart Goossens	Ghent University, Belgium
Mark Hedley	CSIRO ICT Centre, Australia
Dimitris Iakovidis	University of Athens, Greece
Tianzi Jiang	The Chinese Academy of Sciences, China
Ljubomir Jovanov	Ghent University, Belgium
Arto Kaarna	Lappeenranta University of Technology, Finland
Jinman Kang	University of Southern California, USA
Richard Kleihorst	VITO, Belgium
Nikos Komodakis	University of Crete, Crete
Peter Kovesi	University of Western Australia, Australia
Hideo Kuroda	FPT University, Vietnam
Nojun Kwak	Ajou University, Republic of Korea
Florent Lafarge	INRIA, France
Olivier Laligant	IUT Le Creusot, France
Kenneth Lam	The Hong Kong Polytechnic University, China
Patrick Lambert	Polytech' Savoie, France

Peter Lambert	Ghent University, Belgium
Alessandro Ledda	Artesis University College, Belgium
Maylor Leung	Nanyang Technological University, Singapore
Yue Li	CSIRO ICT Centre, Australia
Stefaan Lippens	Ghent University, Belgium
Brian Lovell	University of Queensland, Australia
Guojun Lu	Monash University, Australia
Hiep Luong	Ghent University, Belgium
Anthony Maeder	University of Western Sydney, Australia
Guido Manfredi	Université de Sherbrooke, Canada
Jiri Matas	Czech Technical University, Czech Republic
Gérard Medioni	USC/IRIS, USA
Fabrice Mériaudeau	IUT Le Creusot, France
Alfred Mertins	Universität zu Lübeck, Germany
Jean Meunier	Université de Montréal, Canada
Amar Mitiche	INRS, Canada
Jean-Michel Morel	ENS, France
Yael Moses	The Interdisciplinary Centre, Israel
Adrian Munteanu	Vrije Universiteit Brussel, Belgium
Mai Nguyen-Verger	ENSEA, France
Frank Nielsen	Ecole Polytechnique - Sony CSL, France
Mark Nixon	University of Southampton, UK
Nicolas Paparoditis	IGN, France
Fernando Pereira	Instituto Superior Técnico, Portugal
Stuart Perry	Canon Information Systems Research Australia, Australia
Dijana Petrovska	SudParis, France
Sylvie Philipp-Foliguet	ETIS, France
Wilfried Philips	Ghent University - IBBT, Belgium
Massimo Piccardi	University of Technology Sydney, Australia
Aleksandra Pizurica	Ghent University - IBBT, Belgium
Ljiljana Platisa	Ghent University, Belgium
Dan Popescu	CSIRO, Australia
William Puech	LIRMM, France
Gianni Ramponi	Trieste University, Italy
Paolo Remagnino	Faculty of Technology, Kingston University, UK
Marinette Revenu	ENSICAEN, France
Filip Rooms	Ghent University - IBBT, Belgium
Céline Roudet	Le2i Lab, France
Su Ruan	Université de Reims, France
Luis Salgado Alvarez de Sotomayor	Universidad Politécnica de Madrid, Spain

Paul Scheunders	University of Antwerp, Belgium
Guna Seetharaman	AFRL, USA
Jan Sijbers	University of Antwerp, Belgium
Andrzej Sluzek	Nanyang Technological University, Singapore
Dirk Stroobandt	Ghent University, Belgium
Changming Sun	CSIRO, CMIS, Australia
Hugues Talbot	ESIEE, France
Linda Tessens	Ghent University - IBBT, Belgium
Celine Thillou	UMONS, Belgium
Federico Tombari	University of Bologna, Italy
Frederic Truchetet	Université de Bourgogne, France
Marc Van Droogenbroeck	University of Liège, Belgium
Peter Van Hese	Ghent University, Belgium
Peter Veelaert	University College Ghent, Belgium
Pierre-Henri Wuillemin	UPMC, France
Gerald Zauner	Fachhochschule Oberösterreich, Austria
Pavel Zencik	Brno University of Technology, Czech Republic
Djemel Ziou	Sherbrooke University, Canada

Table of Contents – Part II

Video Processing

Video Quality Analysis for Concert Video Mashup Generation	1
<i>Prarthana Shrestha, Hans Weda, Mauro Barbieri, and Peter H.N. de With</i>	
Speeding Up Structure from Motion on Large Scenes Using Parallelizable Partitions	13
<i>Koen Douterloigne, Sidharta Gautama, and Wilfried Philips</i>	
Mapping GOPS in an Improved DVC to H.264 Video Transcoder	22
<i>Alberto Corrales-García, Gerardo Fernandez-Escribano, and Francisco Jose Quiles</i>	
Scalable H.264 Wireless Video Transmission over MIMO-OFDM Channels	34
<i>Manu Bansal, Mohammad Jubran, and Lisimachos P. Kondi</i>	
A GPU-Accelerated Real-Time NLMMeans Algorithm for Denoising Color Video Sequences	46
<i>Bart Goossens, Hiêp Luong, Jan Aelterman, Aleksandra Pižurica, and Wilfried Philips</i>	
An Efficient Mode Decision Algorithm for Combined Scalable Video Coding	58
<i>Tae-Jung Kim, Bo-Seok Seo, and Jae-Won Suh</i>	
A Novel Rate Control Method for H.264/AVC Based on Frame Complexity and Importance	69
<i>Haibing Chen, Mei Yu, Feng Shao, Zongju Peng, Fucui Li, and Gangyi Jiang</i>	
Digital Image Tamper Detection Based on Multimodal Fusion of Residue Features	79
<i>Girija Chetty, Julian Goodwin, and Monica Singh</i>	

Surveillance and Camera Networks

Fire Detection in Color Images Using Markov Random Fields	88
<i>David Van Hamme, Peter Veelaert, Wilfried Philips, and Kristof Teelen</i>	

A Virtual Curtain for the Detection of Humans and Access Control	98
<i>Olivier Barnich, Sébastien Piérard, and Marc Van Droogenbroeck</i>	
A New System for Event Detection from Video Surveillance Sequences	110
<i>Ali Wali, Najib Ben Aoun, Hichem Karray, Chokri Ben Amar, and Adel M. Alimi</i>	
Evaluation of Human Detection Algorithms in Image Sequences	121
<i>Yannick Benezeth, Baptiste Hemery, Hélène Laurent, Bruno Emile, and Christophe Rosenberger</i>	
Recognizing Objects in Smart Homes Based on Human Interaction	131
<i>Chen Wu and Hamid Aghajan</i>	
Football Players Classification in a Multi-camera Environment	143
<i>Pier Luigi Mazzeo, Paolo Spagnolo, Marco Leo, and Tiziana D’Orazio</i>	
SUNAR: Surveillance Network Augmented by Retrieval	155
<i>Petr Chmelar, Ales Lanik, and Jozef Mlich</i>	
Object Tracking over Multiple Uncalibrated Cameras Using Visual, Spatial and Temporal Similarities	167
<i>Daniel Wedge, Adele F. Scott, Zhonghua Ma, and Jeroen Vendrig</i>	

Machine Vision

A Template Matching and Ellipse Modeling Approach to Detecting Lane Markers	179
<i>Amol Borkar, Monson Hayes, and Mark T. Smith</i>	
An Analysis of the Road Signs Classification Based on the Higher-Order Singular Value Decomposition of the Deformable Pattern Tensors	191
<i>Bogusław Cyganek</i>	
An Effective Rigidity Constraint for Improving RANSAC in Homography Estimation	203
<i>David Monnin, Etienne Bieber, Gwenaël Schmitt, and Armin Schneider</i>	
Exploiting Neighbors for Faster Scanning Window Detection in Images	215
<i>Pavel Zemčík, Michal Hradiš, and Adam Herout</i>	

Remote Sensing

Optimisation-Based Image Grid Smoothing for SST Images	227
<i>Guillaume Noel, Karim Djouani, and Yskandar Hamam</i>	

Estimating 3D Polyhedral Building Models by Registering Aerial Images	239
<i>Fadi Dornaika and Karim Hammoudi</i>	
Content-Based Retrieval of Aurora Images Based on the Hierarchical Representation	249
<i>Soo K. Kim and Heggere S. Ranganath</i>	
Improved Grouping and Noise Cancellation for Automatic Lossy Compression of AVIRIS Images	261
<i>Nikolay Ponomarenko, Vladimir Lukin, Mikhail Zriakhov, and Arto Kaarna</i>	
New Saliency Point Detection and Evaluation Methods for Finding Structural Differences in Remote Sensing Images of Long Time-Span Samples	272
<i>Andrea Kovacs and Tamas Sziranyi</i>	
Recognition, Classification and Tracking	
Regularized Kernel Locality Preserving Discriminant Analysis for Face Recognition	284
<i>Xiaohua Gu, Weiguo Gong, Liping Yang, and Weihong Li</i>	
An Appearance-Based Prior for Hand Tracking	292
<i>Mathias Kölsch</i>	
Image Recognition through Incremental Discriminative Common Vectors	304
<i>Katerine Díaz-Chito, Francesc J. Ferri, and Wladimiro Díaz-Villanueva</i>	
Dynamic Facial Expression Recognition Using Boosted Component-Based Spatiotemporal Features and Multi-classifier Fusion	312
<i>Xiaohua Huang, Guoying Zhao, Matti Pietikäinen, and Wenming Zheng</i>	
Gender Classification on Real-Life Faces	323
<i>Cai Feng Shan</i>	
Face Recognition Using Contourlet Transform and Multidirectional Illumination from a Computer Screen	332
<i>Ajmal Mian</i>	
Shape and Texture Based Plant Leaf Classification	345
<i>Thibaut Beghin, James S. Cope, Paolo Remagnino, and Sarah Barman</i>	

A New Approach of GPU Accelerated Visual Tracking	354
<i>Chuantao Zang and Koichi Hashimoto</i>	
Recognizing Human Actions by Using Spatio-temporal Motion Descriptors	366
<i>Ákos Utasi and Andrea Kovács</i>	
Author Index	377

Table of Contents – Part I

Image Processing and Analysis

A Criterion of Noisy Images Quality	1
<i>Sergey V. Sai, Ilya S. Sai, and Nikolay Yu. Sorokin</i>	
Subjective Evaluation of Image Quality Measures for White Noise Distorted Images	10
<i>Atif Bin Mansoor and Adeel Anwar</i>	
Real-Time Retrieval of Near-Duplicate Fragments in Images and Video-Clips	18
<i>Andrzej Śluzek and Mariusz Paradowski</i>	
Toward the Detection of Urban Infrastructure’s Edge Shadows	30
<i>Cesar Isaza, Joaquín Salas, and Bogdan Raducanu</i>	
Neural Image Thresholding Using SIFT: A Comparative Study	38
<i>Ahmed A. Othman and Hamid R. Tizhoosh</i>	
Statistical Rail Surface Classification Based on 2D and $2^{1/2}$ D Image Analysis	50
<i>Reinhold Huber-Mörk, Michael Nölle, Andreas Oberhauser, and Edgar Fischmeister</i>	
Salient-SIFT for Image Retrieval	62
<i>Zhen Liang, Hong Fu, Zheru Chi, and Dagan Feng</i>	
Combined Retrieval Strategies for Images with and without Distinct Objects	72
<i>Hong Fu, Zheru Chi, and Dagan Feng</i>	
Spectral Matching Functions and Ellipse Mappings in Search for More Uniform Chromaticity and Color Spaces	80
<i>Maryam Pahjehfouladgaran and Arto Kaarna</i>	
Anatomy-Based Registration of Isometrically Transformed Surfaces Using Geodesic Area Functionals	93
<i>Boaz Vigdor and Joseph M. Francos</i>	
Trabecular Bone Anisotropy Characterization Using 1D Local Binary Patterns	105
<i>Lotfi Houam, Adel Hafiane, Rachid Jennane, Abdelhani Boukrouche, and Eric Lespessailles</i>	

Segmentation and Edge Detection

Watershed Based Document Image Analysis	114
<i>Pasha Shadkami and Nicolas Bonnier</i>	
A Fast External Force Field for Parametric Active Contour Segmentation	125
<i>Jonas De Vylder, Koen Douterloigne, and Wilfried Philips</i>	
The Extraction of Venation from Leaf Images by Evolved Vein Classifiers and Ant Colony Algorithms	135
<i>James S. Cope, Paolo Remagnino, Sarah Barman, and Paul Wilkin</i>	
Segmentation of Inter-neurons in Three Dimensional Brain Imagery	145
<i>Gervase Tuxworth, Adrian Meedeniya, and Michael Blumenstein</i>	
Noise-Robust Method for Image Segmentation	153
<i>Ivana Despotović, Vedran Jelača, Ewout Vansteenkiste, and Wilfried Philips</i>	
High Definition Feature Map for GVF Snake by Using Harris Function	163
<i>Andrea Kovacs and Tamas Sziranyi</i>	
Adaptive Constructive Polynomial Fitting	173
<i>Francis Deboeverie, Kristof Teelen, Peter Veelaert, and Wilfried Philips</i>	
Long-Range Inhibition in Reaction-Diffusion Algorithms Designed for Edge Detection and Stereo Disparity Detection	185
<i>Atsushi Nomura, Makoto Ichikawa, Koichi Okada, and Hidetoshi Miike</i>	
An Edge-Sensing Universal Demosaicing Algorithm	197
<i>Alain Horé and Djemel Ziou</i>	
A New Perceptual Edge Detector in Color Images	209
<i>Philippe Montesinos and Baptiste Magnier</i>	
Combining Geometric Edge Detectors for Feature Detection	221
<i>Michaël Heyvaert, David Van Hamme, Jonas Coppens, and Peter Veelaert</i>	
Canny Edge Detection Using Bilateral Filter on Real Hexagonal Structure	233
<i>Xiangjian He, Daming Wei, Kin-Man Lam, Jianmin Li, Lin Wang, Wenjing Jia, and Qiang Wu</i>	
Automated Segmentation of Endoscopic Images Based on Local Shape-Adaptive Filtering and Color Descriptors	245
<i>Artur Klepaczko and Piotr Szczypiński</i>	

3D and Depth

Dense Stereo Matching from Separated Views of Wide-Baseline Images	255
<i>Qian Zhang and King Ngi Ngan</i>	
Modeling Wavelet Coefficients for Wavelet Subdivision Transforms of 3D Meshes	267
<i>Shahid M. Satti, Leon Denis, Adrian Munteanu, Jan Cornelis, and Peter Schelkens</i>	
3D Surface Reconstruction Using Structured Circular Light Patterns ...	279
<i>Deokwoo Lee and Hamid Krim</i>	
Computing Saliency Map from Spatial Information in Point Cloud Data	290
<i>Oytun Akman and Pieter Jonker</i>	
A Practical Approach for Calibration of Omnidirectional Stereo Cameras	300
<i>Kang-San Lee, Hyun-Soo Kang, and Hamid Gholamhosseini</i>	
Surface Reconstruction of Wear in Carpets by Using a Wavelet Edge Detector	309
<i>Sergio Alejandro Orjuela Vargas, Benhur Ortiz Jaramillo, Simon De Meulemeester, Julio Cesar Garcia Alvarez, Filip Rooms, Aleksandra Pizurica, and Wilfried Philips</i>	
Augmented Reality with Human Body Interaction Based on Monocular 3D Pose Estimation	321
<i>Huei-Yung Lin and Ting-Wen Chen</i>	
Fusing Large Volumes of Range and Image Data for Accurate Description of Realistic 3D Scenes.....	332
<i>Yuk Hin Chan, Patrice Delmas, Georgy Gimel'farb, and Robert Valkenburg</i>	
Design of a Real-Time Embedded Stereo Smart Camera	344
<i>Frantz Pelissier and François Berry</i>	
Optimal Trajectory Space Finding for Nonrigid Structure from Motion	357
<i>Yuanqi Su, Yuehu Liu, and Yang Yang</i>	
Fast Depth Saliency from Stereo for Region-Based Artificial Visual Attention	367
<i>Muhammad Zaheer Aziz and Bärbel Mertsching</i>	

Algorithms and Optimisations

A Caustic Approach of Panoramic Image Analysis	379
<i>Siyuan Zhang and Emmanuel Zenou</i>	
Projection Selection Algorithms for Discrete Tomography	390
<i>László Varga, Péter Balázs, and Antal Nagy</i>	
Fast Mean Shift Algorithm Based on Discretisation and Interpolation . . .	402
<i>Eduard Sojka, Jan Gaura, Tomáš Fabián, and Michal Krumník</i>	
Learning to Adapt: A Method for Automatic Tuning of Algorithm Parameters	414
<i>Jamie Sherrah</i>	
Pseudo-morphological Image Diffusion Using the Counter-Harmonic Paradigm	426
<i>Jesús Angulo</i>	
Non-maximum Suppression Using Fewer than Two Comparisons per Pixel	438
<i>Tuan Q. Pham</i>	
Hit-or-Miss Transform in Multivariate Images	452
<i>Santiago Velasco-Forero and Jesús Angulo</i>	
Topological SLAM Using Omnidirectional Images: Merging Feature Detectors and Graph-Matching	464
<i>Anna Romero and Miguel Cazorla</i>	
Constraint Optimisation for Robust Image Matching with Inhomogeneous Photometric Variations and Affine Noise	476
<i>Al Shorin, Georgy Gimel'farb, Patrice Delmas, and Patricia Riddle</i>	
Author Index	489

Video Quality Analysis for Concert Video Mashup Generation

Prarthana Shrestha¹, Hans Weda², Mauro Barbieri², and Peter H.N. de With¹

¹ Eindhoven University of Technology
Den Dolech 2, 5600MB, Eindhoven, The Netherlands
{P.Shrestha,P.H.N.de.With}@TUE.nl

² Philips Research Europe*
High Tech Campus 34, 5656AE, Eindhoven, The Netherlands
{Hans.Weda, Mauro.Barbieri}@Philips.com

Abstract. Videos recorded by the audience in a concert provide natural and lively views from different angles. However, such recordings are generally incomplete and suffer from low signal quality due to poor lighting conditions and use of hand-held cameras. It is our objective to create an enriched video stream by combining high-quality segments from multiple recordings, called *mashup*. In this paper, we describe techniques for quality measurements of video, such as blockiness, blurriness, shakiness and brightness. These measured values are merged into an overall quality metric that is applied to select high-quality segments in generating mashups. We compare our mashups, generated using the quality metric for segment selection, with manually and randomly created mashups. The results of a subjective evaluation show that the perceived qualities of our mashups and the manual mashups are comparable, while they are both significantly higher than the random mashups.

Keywords: no-reference video quality, multiple camera, synchronization, blockiness, blurriness, shakiness, brightness.

1 Introduction

It has become a common practice that audiences at musical concerts record videos using camcorders, mobile phones, digital-still cameras, etc. Consequently, several recordings are made simultaneously of the same event. An obvious example is found in YouTube, where the search phrase “nothing else matters london metallica 2009” returns 18 recordings from different users (search date 08-08-2009). We call a set of such recordings captured in an event around the same time a *multiple-camera recording*.

In professional video productions, a multiple-camera recording is used to compose an enriched video by synchronizing the recordings in a common time-line and then selecting the most desirable segments from them. It is our objective

* The work was carried out at Philips Research Europe, Eindhoven, with partial funding of the Dutch BSIK research program MultimediaN.

to automatically generate such a combined video, called *mashup*, with a high-quality content. A multiple-camera recording of a concert provides different viewing angles from the eyes of the audience, creating a lively experience. However, the qualities of these videos are inconsistent and often low as they are recorded usually by non-professionals using hand-held cameras.

We start with a system based on our earlier work that automatically synchronizes a multiple-camera recording, described in [1]. The synchronization is necessary for seamless continuity between the consecutive audio-visual segments. We use the audio features: fingerprints and onsets to find synchronization offsets among the recordings. The idea is that during a concert, multiple cameras record the same audio at least for a short duration even though they might be pointing at different objects. The method requires a minimum of 3 seconds of common audio between the recordings. It is robust against signal degradations and computes synchronization offsets with a high precision of 11.6 ms. We ensure, also manually, that all the recordings used in mashups generation are accurately synchronized.

In this paper, we describe a method for evaluating video signal quality, then selecting high-quality segments in order to facilitate automated generation of mashups. To this end, we identify different factors describing video quality, such as blockiness and shakiness. We measure these factors applying different content analysis techniques and compute the final quality by combining the measured factor values. The quality measurement is performed and tested in the mashups generated from non-professional multiple-camera concert recordings from YouTube.

2 Video Quality Analysis

The quality metrics known from video compression like mean square error or peak signal to noise ratio are not applicable to our problem statement. This is because there is no information available about the actual scene or the camera settings that can be used as a reference for estimating the signal quality. Therefore, we employ a *no-reference*, also called *blind* quality assessment method, which estimates the image quality based on objective measures of different features that influence the perception of quality.

Prior works on no-reference image quality estimation are done in different contexts such as removing artifacts in home videos [2], developing perceptual quality models [3,4], summarizing home videos [5] and estimating network performance in a real-time video transmission [6]. In [2] the lighting and shaking artifacts in home videos are first detected, measured and then removed. The quality of a JPEG compressed image is estimated in [4] according to the blockiness and blurriness measured in the image, while in [3] according to the edge sharpness, random noise level, ringing artifacts and blockiness. In [5], quality of a home video is measured according to spatial features: infidelity, brightness, blurriness, orientation and temporal features: jerkiness, instability. The features are measured not in every frame but in a temporal video segment. In [6], video quality is measured based on the spatial distortions and temporal activities along the frames.

Since there are multiple set of features that can be applicable in the blind quality analysis of concert video recordings, we conduct a pilot test to evaluate some popular quality features, such as *noise*, *motion*, *brightness*. The evaluation is done by two video professionals, who have been working in the research and development of image and video quality tools for more than 5 years. We show them 4 concert videos obtained from YouTube and 12 representative frames and ask for the factors and their level of influence on the perception of quality. As a result, we select the following quality factors: *blockiness*, *blurriness*, *brightness* and *shakiness*. These metrics address the shortcomings of mobile-handheld non-professional cameras, which typically have small lenses, low-cost sensors and embedded compression with limited quality. The influence of *noise* on the test frames is perceived as minimal because all the test recordings were captured digitally. Similarly, we develop methods for measuring the individual quality factors that comply with the results of the pilot test. The following sections describe the methods for measuring the specified factors and computing an overall quality for each frame.

2.1 Blockiness

Blockiness artifacts are a major source of distortion in videos. It is caused by the codecs, such as such as MPEG, JPEG, and H.264 that involve segmenting a frame into non-overlapping blocks, typically containing 8×8 pixels, and quantizing these blocks separately.

Existing methods for blockiness measurement are based on the degree of discontinuity or strength of the edges at the block boundaries (typically, every 8th horizontal and vertical pixel of an image). In [4] blockiness is measured based on the difference in luminance and signal activity across the block boundaries. If the difference is high in luminance and low in signal activity, then the boundary pixel is considered as blocky. In [7] the discontinuity is measured by the luminance variation in block boundaries of the DC component of an image. Then two thresholds, T_{high} and T_{low} are used to measure the strength of the discontinuity, such that if the discontinuity above T_{high} , the boundary pixel is considered a real edge, called hard edge, of the image and if the discontinuity is below T_{high} but above T_{low} the boundary pixel is considered a soft edge, which is the effect of blockiness

The method proposed in [7] requires computing the DCT for every video frame. Considering the high computational cost of DCT, we decide not to apply the method for our multiple-camera recordings. We tested the method proposed in [4] on the concert video frames obtained from YouTube. The results did not correspond to the perceived level of blockiness in our pilot test. The failure to measure the blockiness in the test frames is maybe due to the low visual quality of the test images such that the signal activity measure does not provide any reliable information or perhaps many hard edges are miscalculated as being the effect of blockiness.

We apply an algorithm for blockiness measurement based on the strength of an edge pixel at the block boundary, inspired by [4] and [7]. We operate on

the luminance component, derived from the YCbCr color-space, of the frame as it contains most of the blockiness information. For both the horizontal and the vertical directions, we use the Sobel operator to obtain the gradient image. Then we apply two thresholds, T_h and T_l , to the boundary pixels such that if the gradient value is above T_h , the pixel is considered as a real or hard edge, while if the value is between T_h and T_l the pixel is considered as a soft edge, which causes the blockiness effect. The values of T_l and T_h are chosen as 50 and 150, respectively based on their performance on the test data-set. A block boundary β_{ij} is considered blocky if more than 75% (6 out of 8) boundary pixels correspond to soft edges. Blockiness measurement in the horizontal direction B_h is specified by:

$$B_h = \frac{1}{([\!W/8\!] - 1)([\!H/8\!] - 1)} \sum_{i=1}^{[\!H/8\!]-1} \sum_{j=1}^{[\!W/8\!]-1} \beta_{ij}, \quad (1)$$

where $\beta_{ij} = 1$ if a block boundary is a soft edge and zero otherwise, W and H represent the number of rows and columns of the frame, respectively. The vertical blockiness B_v is computed in a similar way. The blockiness of a frame B is computed as the average of both horizontal and vertical blockiness as $B = (B_h + B_v)/2$. Fig. 1 shows three frames with different amounts of blockiness.

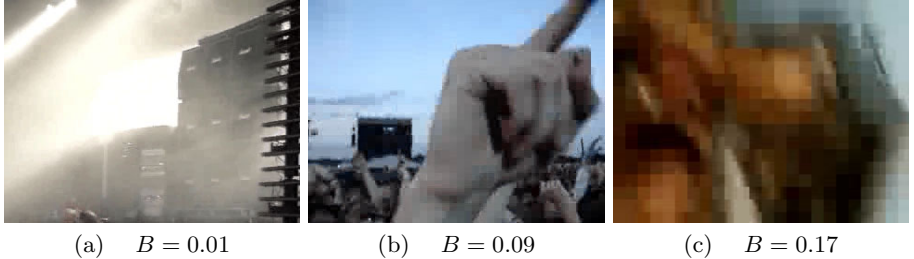


Fig. 1. Examples of test frames with (a) low, (b) medium and (c) high blockiness. The measured blockiness is given by B .

2.2 Blurriness

Blurriness is characterized by the reduction of sharpness of edges. Blur can be caused by lens out of focus or shakiness during capturing, coarse quantization during compression and filtering for blockiness or noise removal during decoding.

We test methods for blurriness measurements based on features: signal activity [4] and average spread of edges [8]. We select the latter method since the blurriness measurement matches better according to our pilot test results. In this method, first the edges are detected using the Sobel operator and the number of pixels contained within the non-zero part of the gradient waveform are counted. This number is specified as the spread of an edge. More specifically, if there are m pixels representing an edge and s pixels representing the total edge spread, then the blurriness score (Z) is calculated as:

$$Z = \begin{cases} s/m & : \text{if } m \neq 0 \\ 0 & : \text{otherwise.} \end{cases} \quad (2)$$

Fig. 2 shows three frames with different amount of blurriness.

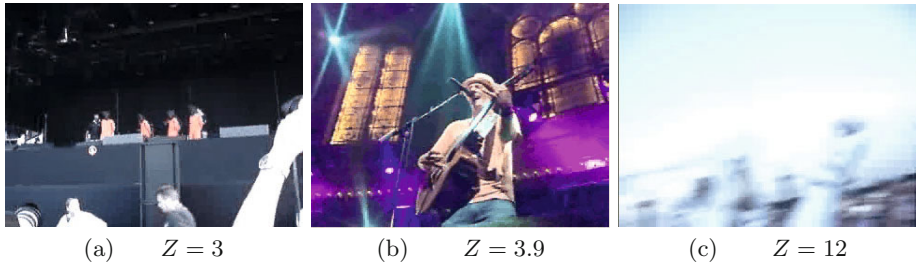


Fig. 2. Example of test frames with (a) low, (b) medium and (c) high blur. The measured blurriness is given by Z .

2.3 Brightness

In video terms, brightness is related to the visual perception of the amount of light coming from the display. Therefore, this parameter depends on the *luminance* of the scene setting, the *contrast* of the involved in video processing and the setting of the display. In a typical indoor concert, the bright areas in a scene are mainly focused towards the stage and the rest of the scene is poorly lit. In our observation of concert recordings, another influencing factor is the amount of *burned pixels*. Burned pixels represent the pixel values clipped by the maximum and minimum luminance values 255 and 0, respectively, caused by a very bright light source against a camera or very dark scenes. Frames containing a higher amount of burned pictures are generally undesirable as they provide little color or texture information and produce a very disturbing effect. Frames with a higher luminance (within the range) and sufficient contrast values are associated with good visibility, pleasant to watch, and colorful images. Therefore, we define brightness here not in the usual video terms but as a function of the three factors mentioned above.

If Y is the luminance component of a frame represented in YCbCr colorspace, the average luminance I_l , contrast I_c and amount of burned pixels I_p of a frame are given by:

$$I_l = \frac{1}{W \times H} \sum_{i=1}^W \sum_{j=1}^H Y(i, j), \quad (3)$$

$$I_c = \sqrt{\frac{1}{W \times H} \sum_{i=1}^W \sum_{j=1}^H (Y(i, j) - I_l)^2}, \quad (4)$$

$$I_p = \max \left(0.1, \frac{1}{W \times H} \sum_{i=1}^W \sum_{j=1}^H p(i, j) \right), \quad (5)$$

$$\text{where, } p(i, j) = \begin{cases} 0 & \text{if } 1 \leq Y(i, j) \leq 254 \\ 1 & \text{if } Y(i, j) = 0 \text{ or } 255. \end{cases} \quad (6)$$

I_p is set to minimal value of 0.1 to allow a limited amount of burned pixels that may occur in any image without resulting in disturbing effects. We compute our definition of the brightness I as: $I = (I_l + I_c)/I_p$. Fig. 3 shows the luminance, contrast, and amount of burned pixels in three example frames.



(a) $I = 2.89$, $I_l = 0.16$, $I_c = 0.55$, $I_p = 0.24$. (b) $I = 1406$, $I_l = 84.17$, $I_c = 56.48$, $I_p = 0.10$. (c) $I = 2627$, $I_l = 235.8$, $I_c = 26.89$, $I_p = 0.10$.

Fig. 3. Example test frames with (a) low, (b) medium and (c) high brightness. The measured values are: brightness (I), luminance (I_l), contrast (I_c) and amount of burned pixels (I_p).

2.4 Shakiness

Shakiness in a video is caused by the instability of a camera, such as when a camera man walks or applies fast zooming or panning operations. Such actions induce motion in unwanted directions, which adversely affects the video quality. In order to measure shakiness in a video, we used the method described in [9].

The camera speed in the horizontal, i.e. *pan*, and vertical direction, i.e. *tilt*, is measured using a luminance projection method [10]. In this method, the luminance values of every row are summed up in a vertical projection and of every column in a horizontal projection. If the camera is moved vertically or horizontally, the corresponding projections will also shift in the same direction. For example, if there is a panning in the right direction, the values of the horizontal projection will shift towards right. The camera motion, pan and tilt, is calculated by correlating the projections along the frames. The speed of the camera is measured in screens per minute, where one screen is equivalent to the horizontal dimension of the frame in case of pan and to the vertical dimension of the frame in case of tilt. The high-frequency components of the camera speed are the result of shakiness, while the low-frequency components are the results of intended camera motion. The amount of shakiness is given by the difference in the pan and tilt speeds before and after applying a low-pass FIR filter (25 tabs). If pan, tilt values before and after filtering is represented by p , t and p_f , t_f , respectively, then for each frame the shakiness measure J is given by:

$$J = \sqrt{(p - p_f)^2 + (t - t_f)^2}. \quad (7)$$

Fig. 4 shows a pan speed before and after the low-pass filtering represented by a thin line and a bold line, respectively.

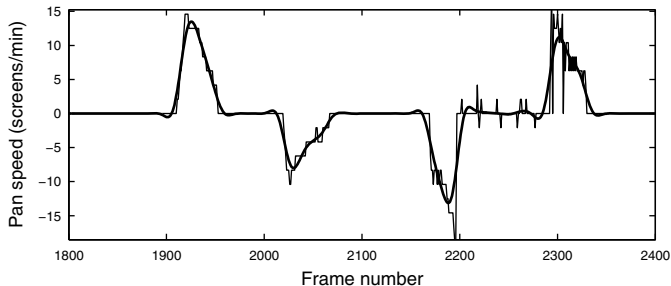


Fig. 4. Camera panning speed in a video before filtering (p), represented by a thin line and after filtering (p_f), represented by a bold line

2.5 Overall Image Quality

The measured values of the quality factors: blockiness, blurriness, brightness, and shakiness determine the quality of a video frame. Since the values corresponding to the different factors have different ranges, we normalize them with respect to their corresponding maximum values obtained from test recordings. After normalization, the values of the factors are in the range $[0,1]$. The values obtained from the method are optimized for the test-set. In ideal case, the values is chosen based on the experimental results with thousands of concert video in different conditions, such as indoor, outdoor, various camera types. However, due to practical limitations we focused on the videos obtained from YouTube.

There are different possible methods to combine the factors to estimate the image quality, such as linear addition [3], linear multiplication [4] and non-linear fusion [5]. No consensus is found in the prior works [3]-[5] over the weights of different quality factors on the overall quality measurement of an image. Based on our observation of the concert recordings and the pilot test, we consider that all the quality factors are equally important. We test the linear addition and multiplication methods on overall quality score. Since the quality score using multiplication method has a wider range than addition method, it is more suitable for comparing the scores of different recordings in a multiple-camera recording. Therefore to compute the overall quality score of a frame, we use the product of the different quality scores. Since shakiness, blurriness and blockiness attribute a negative quality, the factor values are subtracted from one. The image quality score q of a video frame is given by:

$$q = I' (1 - B') (1 - Z') (1 - J'), \quad (8)$$

where I' , B' , Z' and J' represent the normalized values of the brightness, blockiness, blur and shakiness, respectively.

3 Results

Fig. 5 shows the variation of quality scores of three synchronized camera recordings. Visualization of frames at different points in time, indicated in Fig. 5, and their corresponding quality scores are presented in Fig. 6. The frames shown in Fig. 6(a), (b) and (c) are captured simultaneously. However, the views and the quality scores are very different due to different camera positions. Fig. 6(d), (e) and (f) show views from the same cameras as frames in Fig. 6(a), (b) and (c), respectively, but captured later in time and thus with different views.

We apply the described video quality analysis on automatic mashup generation. The mashups are generated from synchronized multiple-camera recordings by selecting 3 to 7 seconds long segments. The segment boundaries are determined based on the change in audio-visual content. The consecutive segments in a mashup are selected from different recordings such that they add diversity and high quality in the mashup content. The quality of a segment is computed as the mean of the quality scores of the frames in the segment. The quality of a mashup depends on the performance of our video quality analysis method, such that a poor analysis of video quality would lead to a poor quality mashup. The mashup quality could be objectively validated if the best and the worst-quality mashups would be made available. However, there are no such existing methods that ensure the definition and subsequent creation of such mashups or allow an objective measure of a mashup quality. Therefore, we measure the perceived quality of our mashup, by a subjective test against two other mashups: one generated by a random selection of segments, i.e. *random mashups*, without considering the quality and another mashup generated manually by a professional video-editor, i.e. *manual mashups*.

As a test set we use three multi-cam recordings, which are captured during concerts by non-professionals and shared in YouTube. Each of the multi-cam recordings contained 4 to 5 recordings with both audio and video streams (in color). The duration of the recordings is between 2.4 and 5.6 minutes and their frame rate is of 25 frames per second. The video resolution is 320×240 pixels. The multiple-camera recordings and the mashups used in the test are made available in website¹, where the filenames C#, Naive, First-fit represent concert number, random mashup and mashup generated by our method.

The random and our quality based mashups contain at least one segment from all the given synchronized recordings and each segment is 3 to 7 seconds long. The manual mashups are created by a professional editor. He was asked to create mashups which are high in signal quality and nice to watch without any special effects and temporal manipulations. It took approximately 16 hours to create 3 mashups from the given test set, using commercially available multi-cam editing software. The considerable time and effort required for creating manual mashups forced us to limit the size of the test set.

The subjective test involves 40 individuals, age between 20 and 30. The 9 mashups, generated using 3 methods and 3 concerts, are shown to the subjects

¹ <http://www.youtube.com/AutomaticMashup#p/u>

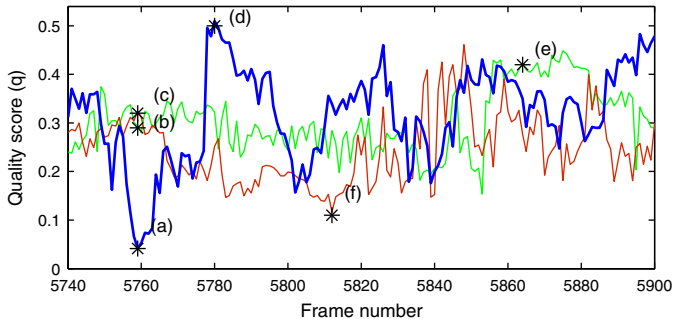


Fig. 5. Quality score of recordings from a three-camera recording, given by different colors, in a common time-line represented by the frame numbers. Frames indicated by ‘*’ are shown in Fig. 6

followed by a questionnaire. We ask the participants to indicate the level of the perceived *image quality* and *camera stability* in a 7 point Likert scale, used extensively in perception tests. The scale is comparable to the *mean opinion score* employed in subjective analysis of the television pictures [11]. The measures image quality and camera stability are chosen from a pilot user-study, as individual quality factors like blurriness and blockiness are difficult to differentiate and evaluate for a general user.

The mean scores of the mashups generated by random, manual and our quality-based methods in terms of perceived camera stability and image quality in Fig. 7(a) and (b), respectively. The confidence intervals are presented graphically as an error bar on the mean score, such that if the test is repeated with other participants from the same target group, there is 95% probability that the mean score will remain within the interval. The scores are further analyzed to verify whether the differences between the mean scores are statistically significant. We apply a two-way analysis of variance (ANOVA) with repeated measures. The *method* (random, quality-based, manual) and *concert* (C1-C3) are treated as within-subject independent variables and the response of the participants are treated as a dependent variable. The results are presented in terms of *F-statistic* and *p-value* such that if $p < 0.05$ there is 95% confidence that the means are significantly different. Since ANOVA indicates if the means are significantly different, but it does not distinguish which means are different, an additional *Tukey post-hoc* test is performed on both independent variables. The results provide pairwise comparisons of the means and their confidence intervals.

As shown in Fig. 7(a), camera stability score of the manual mashups appear to be higher than the mashups generated by other methods in all three concerts. According to the ANOVA analysis, a significant main effect is found for methods ($F = 4.593$, $p = 0.010$) and for concerts ($F = 31.853$, $p < 0.001$). A Tukey test on *method* shows that score of the random mashup is significantly lower than the mashups generated by other methods. Similarly, a Tukey test on *concert* shows that C1 is significantly different from C2 and C3.

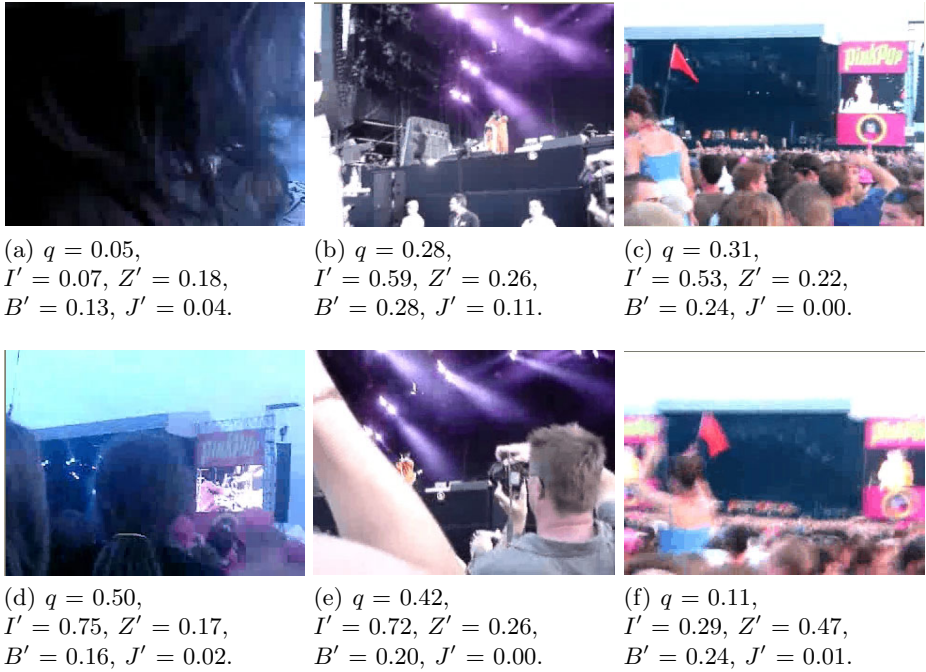


Fig. 6. The quality score and normalized values of brightness (I'), blurriness (Z'), blockiness (B') and shakiness (J') of the frames given by '*' in Fig. 5

As shown in Fig. 7(b), image quality score of the random mashups seems to be lower than the mashups generated by other methods in all three concerts. The mean across all the three concerts shows that random scores lower than the other two methods, which score about the same. From the ANOVA analysis, a significant main effect was found for *method* ($F = 7.833$, $p < 0.001$) and for *concert* ($F = 16.051$, $p < 0.001$). A Tukey test on *method* shows that the quality of random mashups is significantly lower than mashups generated by other two methods. Similarly, a Tukey test on *concert* shows that C3 is significantly different than other concerts.

It is expected, as seen case of C1 and C2, the random mashups are perceived as more shaky and low in image-quality because camera stability and image quality are not taken into account during mashup generation. However, in C3 the quality-based mashup is perceived as shaky and low quality as the random mashups. This could be due to the low visual quality of the camera recordings of concert C3, containing objects in fast motion (dancing) and dynamic lights, which cause errors in the shakiness detection and brightness measurement.

The analysis of the subjective test shows that the image quality and shakiness scores are dependent on the concerts and the methods. The scores of the random mashups, on average across different concerts, are significantly lower than the other two mashups, while the quality-based mashups and the manual mashups

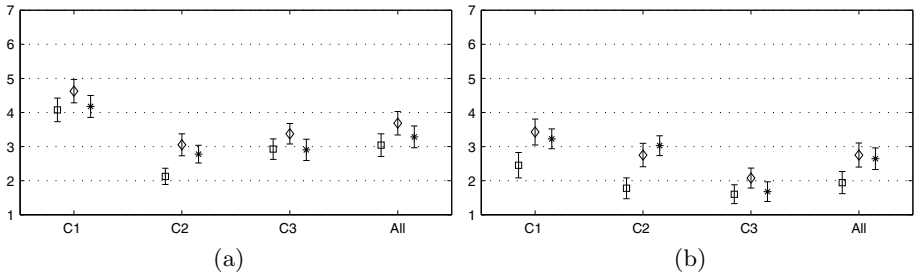


Fig. 7. Mean scores for (a) camera stability and (b) image quality. The methods are represented as: random (\square), manual (\diamond), and quality-based (*). The horizontal axis represents the mashups corresponding to the three concerts C1–C3. “All” represents the mean across all the concerts. Error bars show confidence intervals of 95% of the mean value.

are comparable. However, if the concert recordings contain low image quality the mashup quality cannot be improved.

4 Conclusion

This paper describes a method for evaluating signal quality of concert video recordings captured by the audience. We measure the blockiness, blurriness, brightness and shakiness of the video frames and then combine them into an overall multiplicative measure for the quality of a video segment. The method is applied to automatically select high-quality segments in a mashup from multiple-camera recordings. We compare the quality of these mashups against the mashups created by a random selection of segments and by a professional video-editor. The subjective evaluation shows that the perceived quality of our mashups is comparable to the mashups created by the professional video-editor and significantly higher than the mashups generated randomly. Further analysis shows that the mashup quality depends not only on the methods used on generating them but also on the recording quality of the concert videos.

References

1. Shrestha, P., Weda, H., Barbieri, M., Sekulovski, D.: Synchronization of multiple camera videos using audio-visual features. *IEEE Trans. on Multimedia* 12(1), 79–92 (2010)
2. Yan, W., Kankanhalli, M.S.: Detection and removal of lighting & shaking artifacts in home videos. In: *Proc. of the 10th ACM Int. Conf. on Multimedia*, pp. 107–116 (2002)
3. Li, X.: Blind measurement of blocking artifacts in images. In: *Int. Conf. on Image Processing*, vol. 1, pp. 449–452 (2002)
4. Wang, Z., Sheikh, H.R., Bovik, A.C.: No-reference perceptual quality assessment of JPEG compressed images. In: *Proc. of Int. Conf. on Image Processing*, vol. 1, pp. 477–480 (2002)

5. Mei, T., Zhu, C.-Z., Zhou, H.-Q., Hua, X.-S.: Spatio-temporal quality assessment for home videos. In: Proc. of the 13th ACM Int. Conf. on Multimedia, pp. 439–442 (2005)
6. Yang, F., Wan, S., Chang, Y., Wu, H.R.: A novel objective no-reference metric for digital video quality assessment. *IEEE Signal Processing Letters* 4(10), 685–688 (2005)
7. Gao, W., Mermer, C., Kim, Y.: A de-blocking algorithm and a blockiness metric for highly compressed images. *IEEE Trans. on Circuits and Systems for Video Technology* 12, 1150–1159 (2002)
8. Ong, E., et al.: A no-reference quality metric for measuring image blur. In: Proc. 7th Int. Symp. on Signal Processing and Its Applications, vol. 1, pp. 469–472 (2003)
9. Campanella, M., Weda, H., Barbieri, M.: Edit while watching: home video editing made easy. In: Proc. of the IS&T/SPIE Conf. on Multimedia Content Access: Algorithms and Systems, vol. 6506, pp. 65060–65060 (2007)
10. Uehara, K., Amano, M., Ariti, Y., Kumano, M.: Video shooting navigation system by real-time useful shot discrimination based on video grammar. In: Proc. of the Int. Conf. on Multimedia & Expo., pp. 583–586 (2004)
11. RECOMMENDATION: ITU-R BT.500. Methodology for the subjective assessment of the quality of television pictures (2002)

Speeding Up Structure from Motion on Large Scenes Using Parallelizable Partitions

Koen Douterloigne, Sidharta Gautama, and Wilfried Philips

Department of Telecommunications and Information Processing
(UGent-TELIN-IPI-IBBT)

Ghent University, St-Pietersnieuwstraat 41, B-9000 Ghent, Belgium
koen.douterloigne@telin.ugent.be

Abstract. Structure from motion based 3D reconstruction takes a lot of time for large scenes which consist of thousands of input images. We propose a method that speeds up the reconstruction of large scenes by partitioning it into smaller scenes, and then recombining those. The main benefit here is that each subscene can be optimized in parallel. We present a widely usable subdivision method, and show that the difference between the result after partitioning and recombination, and the state of the art structure from motion reconstruction on the entire scene, is negligible.

Keywords: Structure from motion, 3D reconstruction, speedup, large scenes.

1 Introduction

More and more applications require accurate three dimensional (3D) models of the world, e.g. planning urban environments and infrastructures, automated object detection, or augmented reality and CGI in movies. To create these 3D models various options exist, including mobile mapping, laser scanning, or manual surveying. These ground based acquisition methods all have in common that they are time consuming, especially for larger areas. The most practical approach to quickly cover a lot of terrain is that of aerial imaging, where 2D pictures are captured and then processed to create a 3D model [4]. Just like in [13], the 3D model is derived from multiple pictures of the same area, taken under different angles.

To find the initial position of the pictures usually GPS information is used. A major problem however is that we can not always rely on GPS information being available. In land based mapping anything that interferes with the line of sight to the GPS satellite has a negative effect on the reception, e.g. trees or large buildings. But even low altitude aerial surveillance can not always count on a GPS link, as certain regions have active jamming devices (e.g. conflict zones). A good solution to handle these problems is to employ structure from motion. Instead of relying on the GPS information, the position of the camera is determined from image correspondences. Consecutive aerial pictures have a

high percentage of overlap, making this possible. However the computation time required by structure from motion is approximately quadratic in the amount of points in the scene [15], and so does not scale well to very large scenes.

If we could limit the structure from motion optimization to small scenes, and then later combine all small scenes into one global scene, the downsides of the quadratic behaviour would be largely avoided. Additionally, every small scene can be optimized in parallel, further increasing the reconstruction speed. In this paper we work out the details involved in splitting and recombining a large scene, and evaluate how the final result changes with respect to the original, slow reconstruction. We do not compare with any ground truth, as the absolute accuracy of the 3D model obtained by various reconstruction methods has already been evaluated in [11].

Previous work on speeding up structure from motion for large scenes includes sub-sampling and hierarchical decomposition [10]. While effective, the downside here is that for very large scenes, the required time is still quadratic. The methods presented in [14] and [9] also use partial reconstructions to speed up the final result. However these techniques use the Hessian of the reprojection error and its eigenvector to split up the global scene, which implies that the scene must be already approximately reconstructed. Again, this requires a lot of time for large scenes consisting of many pictures.

The rest of this paper is arranged as follows. First we briefly explain structure from motion and bundle adjustment. Next we present the theory behind our method to speed up the computations, followed by experiments to evaluate the accuracy on practical data. We end with a conclusion.

2 Structure from Motion

2.1 3D Reconstruction with Bundle Adjustment

We first discuss the problem of the 3D reconstruction of a scene, based on multiple 2D views from different locations. Given n 3D points which are observed by m cameras, we can write as \mathbf{x}_{ij} the projection of point i in camera j , with $i = 1..n$ and $j = 1..m$. The projection from a 3D point \mathbf{X} to a 2D point \mathbf{x} can be written compactly in homogeneous coordinates as

$$\lambda \mathbf{x} = \mathbf{M}\mathbf{X} \quad (1)$$

with λ a scale factor and \mathbf{M} the 3 x 4 homogeneous camera matrix, with 11 independent parameters [6]. This projective camera model can be simplified to Euclidean cameras, for which \mathbf{M} can be decomposed into

$$\mathbf{M} = \mathbf{K}[\mathbf{R}|\mathbf{t}] \quad (2)$$

where \mathbf{t} is augmented to \mathbf{R} . Here \mathbf{K} is the 3 x 3 intrinsic calibration matrix containing the camera's optical properties, such as focal length, principal point and aspect ratio. The camera's extrinsic parameters are given by \mathbf{R} , a 3 x 3

rotation matrix, and \mathbf{t} , a 3×1 translation vector. From this we calculate the camera’s position as $-\mathbf{R}^T \mathbf{t}$. Furthermore we can incorporate the optical image distortion caused by imperfect lenses into (1) by writing $\lambda r(\mathbf{x}) = \mathbf{M} \mathbf{X}$, with $r(\cdot)$ the distortion function. This function maps a correct (undistorted) image to its distorted version. e.g. $r(\mathbf{x}) = 1 + k_1 \|\mathbf{x}\|^2 + k_2 \|\mathbf{x}\|^4$. Many more distortion functions exist [2], as well as camera calibration techniques to determine the parameters of the distortion function in advance [5,3].

The 3D reconstruction from multiple views now comes down to finding values for all cameras \mathbf{M} and all 3D points \mathbf{X} so that the difference between the computed position of \mathbf{x} from (1) and the measured position of \mathbf{x} is minimized,

$$\min_{\mathbf{M}_j, \mathbf{X}_i} \sum_{i=1}^n \sum_{j=1}^m d(\mathbf{M}_j \mathbf{X}_i, \mathbf{x}_{ij})^2 \quad (3)$$

with $d(\mathbf{x}, \mathbf{y})$ the Euclidean distance. The summation of all distances is called the *reprojection error*, and (3) is typically minimized using bundle adjustment [15]. Due to the sparse nature of the equations (the parameters of individual 3D points \mathbf{X} and cameras \mathbf{M} do not interact) several optimizations for speed can be applied. Work on this by Lourakis et al. resulted in the open source software package sba [7], using a modified version of the Levenberg-Marquardt algorithm for the iterative optimization. Still, the required time for optimization is at least quadratic in the number of 3D points, although exact timings depend on the scene under consideration [7]. When all matrices \mathbf{M} are known, a dense reconstruction of the scene can be created, using for example the methods described in [4].

2.2 Finding Corresponding Points

The bundle adjustment requires knowledge of points \mathbf{x}_{ij} . In other words, we must identify points in all images that correspond to the same physical location or 3D point. Several algorithms exist that do this, most notably SIFT [8] and SURF [1]. These methods extract feature points that are likely to be recognized in another image, and then match those feature points based on Euclidean distances between their associated feature descriptors. The amount of point matches that are generated, as well as their reliability, depend on the input images and on several parameters in the algorithms. In general, regions without distinctive features will contain less feature points. While this makes the found features more robust, on a large scene it can also give rise to regions without any feature points. This is not desirable, so we tweak the parameters of the feature point detection algorithm to give roughly the same amount of feature points for all input images (e.g. 500 points). If too much features are found, the parameters are tightened, and vice versa.

2.3 Avoiding Local Optima

Due to its nonlinearity, the reprojection error defined by (3) contains many local optima. This problem gets worse when some of the pointmatches found

in [2,2] are not correct. When solving the minimization, one should of course avoid these local optima. The method developed in [12] solves this problem by always starting from an approximate reconstruction of the scene from a previous bundle adjustment iteration, and then adding just a couple of new views to it. Additionally, these new views are first roughly positioned using RANSAC based on the point matches. This incremental approach ensures that an intermediate solution is always close to the global optimum, thus helping the gradient based Levenberg-Marquardt algorithm. The downside is that a lot of computation time is spent on re-optimizing parts of the scene that were already reconstructed, as [3] always considers all points and cameras.

3 Proposed Method

3.1 Splitting the Global Scene

While considering the whole scene in [3] gives the most reliable result, it is clearly not optimal w.r.t. time. We propose a divide and conquer approach, splitting the scene into several overlapping subsets of size S with overlap O . Then each subset is optimized separately, after which the results are combined. Figure 1 shows an example. Ideally a subset consists of cameras positioned close to eachother. The problem is that we generally do not know the positions of the cameras in advance. However, in practice it is often the case that pictures taken close together in time, are also close in space. Thus we subdivide the scene based on the order in which the images were acquired. To keep things manageable, we use a constant S and O . One could think of a dynamic splitting scheme where S and O change based on the quality of feature matches, or closeness of initial image transformations. However due to the large number of possibilities we leave this as future work.

The values of S and O determine the balance between speed and accuracy, where accuracy is defined as the difference between the combined subsets and the result we get without splitting. Smaller subsets require less time to optimize, but are prone to wind up in local minima. Smaller overlaps decrease computation time as well, but also decrease the accuracy of the combined result. The reason for this is that the bundle adjustment can only position cameras in relation to

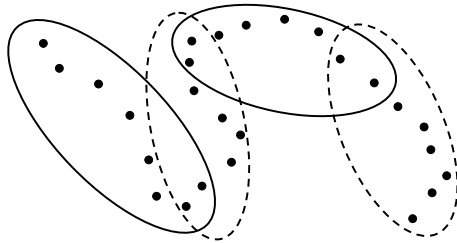


Fig. 1. Splitting a global scene into subscenes, with $S = 8$ and $O = 2$. The black dots are the (unknown) camera positions.

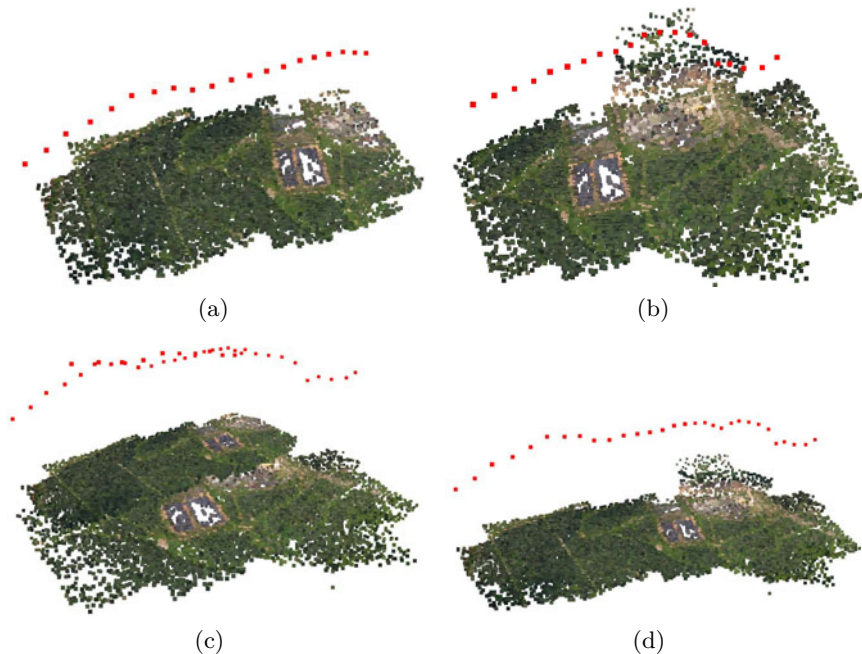


Fig. 2. Combining overlapping scenes with $S = 20$ and $O = 10$, where the input was a set of 10 megapixel aerial images taken from a plane at an altitude of 150m. (a) Scene reconstructed from images 1 to 20. (b) Scene reconstructed from images 11 to 30. (c) Combination without coordinate transformation. (d) Combination after coordinate transformation. The red squares are the reconstructed positions of the cameras (i.e. the plane’s position at each time), and the other dots are reconstructed points of the landscape.

each other, also known as the gauge problem [15], and so does not create a global coordinate system. To combine two or more sets of 3D points in different coordinate systems, we have to know points that are present in both scenes, and based on this correspondence find the coordinate transformation. The more points we know, i.e. the larger the overlap, the closer this computed transformation will be to the actual transformation.

3.2 Combining Two Overlapping Scenes

Combining two overlapping scenes consists of three steps.

1. Locate the points that are visible in both scenes. Obviously the 3D location of the points is not useable for this purpose, as that depends on the coordinate system of the scene. Instead we find matches based on the indices of the feature points. Every view of a 3D point in a certain camera is associated with a feature point \mathbf{x}_{ij} (see section 2). All feature points in a certain camera are indexed. A point \mathbf{p} of scene 1, visible in cameras c_p , matches a point \mathbf{q}

of scene 2, visible in cameras c_q , if there exists a pair of cameras (c_p, c_q) for which the indices of the feature points are the same. Furthermore, the camera pair must be related by the known overlap O , so we can write $c_p = c_q + O$. This search can be executed very fast using a hashtable on the point indices.

- Find the transformation between these sets of points. Given two sets of matching points (x_i, y_i, z_i) and (x'_i, y'_i, z'_i) , we can write

$$\begin{bmatrix} x'_i \\ y'_i \\ z'_i \end{bmatrix} = \begin{bmatrix} T_{11} & T_{12} & T_{13} & T_{14} \\ T_{21} & T_{22} & T_{23} & T_{24} \\ T_{31} & T_{32} & T_{33} & T_{34} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} \quad (4)$$

with T_{ij} the values of the affine coordinate transformation $\mathbf{T} = [\mathbf{A}|\mathbf{B}]$, with \mathbf{A} the affine component and \mathbf{B} the translation. Rewriting (4) gives

$$\begin{bmatrix} x'_1 \\ y'_1 \\ z'_1 \\ \vdots \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 \\ & & & & & & & \vdots & & & & \end{bmatrix} \begin{bmatrix} T_{11} \\ T_{12} \\ \vdots \\ T_{34} \end{bmatrix} \quad (5)$$

which in general is an overdetermined system that can be solved using singular value decomposition. The pseudo-inversion can become slow when the number of matches is large, so we will only use a subset (e.g. 100) of the available matches to determine \mathbf{T} .

- Apply \mathbf{T} to transform the points and the camera positions, and average cameras and points that occur twice due to the overlap. Transforming a point is easy, simply $\mathbf{X}' = \mathbf{A}\mathbf{X} + \mathbf{B}$. However from (2) the position of a camera \mathbf{c} is defined by its rotation \mathbf{R} and translation \mathbf{t} , $\mathbf{c} = -\mathbf{R}^T\mathbf{t}$, so to put the camera in the new coordinate system we must find new values for \mathbf{R}' and \mathbf{t}' so that $\mathbf{c}' = \mathbf{A}\mathbf{c} + \mathbf{B} = -\mathbf{R}'^T\mathbf{t}'$. Expanding this gives

$$\begin{aligned} \mathbf{c}' &= \mathbf{A}\mathbf{c} + \mathbf{B} = \mathbf{A}(-\mathbf{R}^T\mathbf{t}) + \mathbf{B} \\ &= (-\mathbf{A}\mathbf{R}^T\mathbf{N}\mathbf{N}^{-1})\mathbf{t} + \mathbf{B} \\ &= (-\mathbf{A}\mathbf{R}^T\mathbf{N})(\mathbf{N}^{-1}\mathbf{t}) + (-\mathbf{A}\mathbf{R}^T\mathbf{N})(-\mathbf{A}\mathbf{R}^T\mathbf{N})^{-1}\mathbf{B} \\ &= (-\mathbf{A}\mathbf{R}^T\mathbf{N})(\mathbf{N}^{-1}\mathbf{t} + (-\mathbf{A}\mathbf{R}^T\mathbf{N})^{-1}\mathbf{B}) \\ &= -\mathbf{R}'^T\mathbf{t}' \end{aligned} \quad (6)$$

where \mathbf{N} is a 3 x 3 normalization matrix added to ensure that $-\mathbf{A}\mathbf{R}^T\mathbf{N}$ is a true rotation matrix, i.e. the columns of $-\mathbf{A}\mathbf{R}^T\mathbf{N}$ must have norm 1. This means that \mathbf{N} will only have non-zero elements on its diagonal, with N_{ii} , $i = 1..3$ equal to the inverse of the norm of column i of $-\mathbf{A}\mathbf{R}^T$. To summarize, for every camera \mathbf{c}_j , $j = 1..m$ we can find the transformed camera \mathbf{c}'_j by calculating

$$\mathbf{R}'_j = (\mathbf{A}\mathbf{R}_j^T\mathbf{N}_j)^T \quad (7)$$

$$\mathbf{t}'_j = \mathbf{N}_j^{-1}\mathbf{t} + (-\mathbf{A}\mathbf{R}_j^T\mathbf{N}_j)^{-1}\mathbf{B} \quad (8)$$

Finally we also detect and remove some impossible points. A likely scenario is a point that is visible from a camera in scene 1 with a certain feature point index, and visible from the same camera in scene 2 (offset by the overlap O) but with a different feature point index. This indicates that some views of this point are incorrect. Another possibility is a triangular match, where a point in scene 1 matches a point in scene 2 from one camera view, and another point in scene 2 from another camera view. When this happens we remove all involved points from the combined scene, as we can not be sure which points and matches are correct.

4 Evaluation

4.1 Effective Range

Our proposed method is only effective starting from a certain size of the global scene. Due to overhead incurred at the start of a bundle adjustment the method of subdividing can be slower than running the bundle adjustment on the complete scene when this scene is small. Furthermore, for small scenes the incremental addition of images to the reconstructed scene is approximately linear in time, so it makes no sense to split it. Figure 3 illustrates this, showing the time required to reconstruct a scene of 629 images on an Intel Core 2 Duo T9300 CPU. The final scene counts 479,212 points and took over 18 hours to compute. When less than 200 images are included in the incremental optimization, the time required to add more images is relatively small. However when the scene includes more than 400 images, adding additional views and points takes a lot of time. The reason is that the speed of the bundle adjustment is quadratic in the number of 3D points in the scene, which is linked to the number of views. Starting the Levenberg-Marquardt algorithm close to the optimum reduces the amount of iterations, but does not reduce the time required for one iteration. As long as the scene is small, the overhead from starting each bundle adjustment run will be larger than the actual time spent optimizing, resulting in a linear behaviour. By partitioning to sizes that fall inside this linear zone, our method has an approximate complexity of $O(N)$ instead of $O(N^2)$.

4.2 Influence of Subscene Size S

As our method focuses on improving computation time, and does not deal with the accuracy of the result, we want to minimize the difference between point clouds generated by state of the art methods and our proposed method. The logical choice of method to compare to is the output generated by the method from [12]. Table 1 presents an evaluation of the influence of the partitioning size S for a static overlap size $O = 10$. The columns presented are as follows. First the *time subset*, in seconds, is the time required to optimize the slowest (i.e. most difficult) subscene. This is the total running time if the reconstruction would run completely parallel. Next is the *time total*, which is the optimization

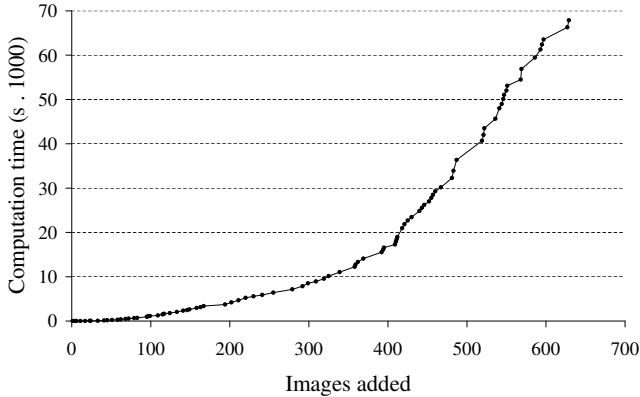


Fig. 3. The time required to optimize a large scene by incrementally adding images, showing the approximately quadratic relation between scene size and calculation speed

Table 1. Evaluation of the influence of the partitioning size S , for $O = 10$. The total time to optimize the scene using the methods from literature [12] was 2829 seconds. All timing results were achieved on an Intel Core 2 Duo T9300 CPU. The total scene size is about $25 \times 20 \times 10$ units. See the text for details.

S	time subset (s)	time total (s)	μ	σ	μ_{opt}	σ_{opt}
20	70	667	0.02315	0.01302	0.01248	0.00921
30	147	685	0.01937	0.01008	0.00863	0.00750
40	221	745	0.03692	0.02054	0.01032	0.00978
50	314	836	0.01655	0.01027	0.01520	0.01112
60	408	1001	0.00841	0.00527	0.00232	0.00241
70	492	1341	0.00742	0.00825	0.00320	0.00419

time required when the method runs sequential, in case only 1 CPU core would be available. Note that for all values of S the total running time is lower than when we would not split the scene. This confirms our observations from figure 3. The next four columns give accuracy results. The first μ and σ are the mean and standard deviation of all the differences between the 3D positions of all points in the reconstructed scene using our method and the method from literature. The second μ_{opt} and σ_{opt} are the results after running a bundle adjustment on the recombined scenes. This fixes deviations introduced by the splitting, but of course at the cost of some computation time. The presented numbers only make sense in relation to the total scene size, which is about $25 \times 20 \times 10$ units. The reported errors are thus a factor 10^4 smaller than the scene size, meaning that the scene reconstructed with our method will be visually identical to the scene reconstructed using state of the art methods.

5 Conclusion

In this paper we have presented a fast and accurate approach to creating 3D models of large scenes using structure from motion. We have shown that our method of partitioning and recombining the global scene performs much faster than the existing state of the art approach, while giving a result that is visually identical. Future work can focus on a dynamic estimation of the partitioning and overlap sizes, based on statistics taken from the point matches.

References

1. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.* 110(3), 346–359 (2008)
2. Claus, D., Fitzgibbon, A.W.: A rational function lens distortion model for general cameras. In: *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, vol. 1, pp. 213–219. IEEE Computer Society, Washington (2005)
3. Douterloigne, K., Gautama, S., Philips, W.: Fully automatic and robust UAV camera calibration using chessboard patterns. In: *IEEE International Geoscience and Remote Sensing Symposium*, pp. 551–554 (July 2009)
4. Furukawa, Y., Ponce, J.: Accurate, Dense, and Robust Multi-View Stereopsis. *IEEE Trans. on Pattern Analysis and Machine Intelligence* (2009)
5. Gennery, D.B.: Generalized camera calibration including fish-eye lenses. *Int. J. Comput. Vision* 68(3), 239–266 (2006)
6. Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York (2003)
7. Lourakis, M.A., Argyros, A.: SBA: A Software Package for Generic Sparse Bundle Adjustment. *ACM Trans. Math. Software* 36(1), 1–30 (2009)
8. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision* 60(2), 91–110 (2004)
9. Ni, K., Steedly, D., Dellaert, F.: Out-of-core bundle adjustment for large-scale 3d reconstruction. In: *IEEE International Conference on Computer Vision*, vol. 0, pp. 1–8 (2007)
10. Nistér, D.: Reconstruction from uncalibrated sequences with a hierarchy of trifocal tensors. In: Vernon, D. (ed.) *ECCV 2000*. LNCS, vol. 1842, pp. 649–663. Springer, Heidelberg (2000)
11. Seitz, S.M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2006*, pp. 519–528. IEEE Computer Society, Washington (2006)
12. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: Exploring image collections in 3d. In: *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2006)* (2006)
13. Snavely, N., Seitz, S.M., Szeliski, R.: Modeling the world from internet photo collections. *Int. J. Comput. Vision* 80(2), 189–210 (2008)
14. Steedly, D., Essa, I., Delleart, F.: Spectral partitioning for structure from motion. In: *Proceedings of the Ninth IEEE International Conference on Computer Vision, ICCV 2003*, p. 996. IEEE Computer Society, Washington (2003)
15. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment - a modern synthesis. In: *Proceedings of the International Workshop on Vision Algorithms, ICCV 1999*, pp. 298–372. Springer, London (2000)

Mapping GOPS in an Improved DVC to H.264 Video Transcoder

Alberto Corrales-Garcia, Gerardo Fernandez-Escribano, and Francisco Jose Quiles

Instituto de Investigación en Informática de Albacete (I3A)
Universidad de Castilla-La Mancha
02071 Albacete, Spain
{albertocorrales,gerardo,paco}@dsi.uclm.es

Abstract. In mobile to mobile video communications, both transmitter and receptor should keep low complexity constrains during video compression and decompression processes. Traditional video codecs have highly complex encoders and less complex decoders whereas the Wyner-Ziv video coding paradigm inverses the complexity by using more complex decoders and less complex encoders. For this reason, transcoding from Wyner-Ziv to H.264 provides a suitable framework where both devices have low complexity constraints. This paper proposes a flexible Wyner-Ziv to H.264 transcoder which allows us to map from a Wyner-Ziv GOP pattern to a I11P H.264 GOP. Furthermore, the transcoding process is improved when reusing the motion vectors that have been calculated during the Wyner-Ziv decoding process to reduce the H.264 motion estimation complexity. Accordingly a time reduction up to 72% is obtained without significant rate-distortion loss.

Keywords: DVC, Wyner-Ziv, H.264, Transcoding.

1 Introduction

The newest generations of mobile communication systems (such as 4G) offer attractive services such as video telephony and video conference to mobile users. These systems provide more advanced types of interactive and distribution services in which video is one of the most prominent applications for these multimedia communications. In this kind of services, both the transmitter and receiver devices may not have sufficient computing power, resources or complexity constraints to perform complex video algorithms (both coding and decoding). By using traditional video codecs such as H.264 [1] these low complexity requirements have not been satisfied because H.264 has high complexity at the encoder side. Therefore, these mobile video communications that employ traditional video codecs lead to an inefficient configuration because the encoders sacrifice Rate – Distortion (RD) performance by using only the lower complexity encoding tools. Distributed Video Coding (DVC) [2] is a more recent video coding paradigm that offers a video coding scheme where the majority of the complexity is moved to the decoder which deals with simpler encoders. Since one year ago, DVC to H.26X transcoders [3][4] have

appeared in the multimedia community as a solution for this mobile to mobile video communication, as it is shown in figure 1. In this framework the majority of the computation is moved to the network where the transcoder is allocated and the simpler algorithms (DVC encoding and H.26X decoding) are implemented in the end user devices. At the moment transcoding architecture offers the most suitable solution for mobile-to-mobile video communications due to the low complexity in both extremes. In the literature the Group of Pictures (GOP) pattern in the DVC architectures is formed by two kinds of frames: Wyner-Ziv (WZ) and Key (K). Normally with GOP sizes ranging from 2, 4 and 8 (WZ frames between two K frames) although other GOP sizes are also allowed. On the other hand, in H.264 the most suitable GOP pattern for mobile communications is labeled as I11P [5] which is formed by one I-frame followed by 11 P-frames and with GOP size of twelve. These GOP sizes / patterns mismatches between DVC and H.264 entail a problem that must be solved in the proposed scenario.



Fig. 1. System using a DVC / H.26X transcoder

At this point, this paper is a straight forward step in the framework of DVC to H.264 video transcoders and offers a GOP mapping solution between kinds of GOP sizes / patterns as well as some refinement in the motion estimation algorithm developed at H.264 encoding algorithm as part of the whole video transcoder to make a faster process. Accelerating the transcoding process is a very important task in order to try to reach a real time communication.

This paper is organized as follows: Section 2 identifies the state-of-the-art in DVC based transcoders. Section 3 shows the proposed mapping algorithm for the DVC to H.264 video transcoder which is evaluated in Section 4 with some simulation results. Finally, in Section 5, conclusions are presented.

2 Related Work

The main task of a transcoder is to convert from a source format to another one; this task must be executed as efficiently as possible though. Consequently, transcoding techniques focus on improving the second stage by using information gathered during the first one. The key issues to manage are the time reduction and the quality – bitrate penalty. Transcoding algorithms between traditional video coding standards are more suitable to be used because both formats keep many features whereas DVC is more different.

Firstly, in 2005 [6] WZ coding was proposed as a candidate in a transcoding scene, however it only introduced the idea and the benefits of this new transcoding paradigm to support low cost video communications but it did not offer a practical implementation. The first WZ transcoder architecture was presented in 2008 by Peixoto et al. in [3]. In this approach the authors designed a WZ to H.263 transcoder to support mobile communications. This transcoder makes a mapping between WZ and H.263 GOPs, including some Motion Estimation (ME) refinement for P and B slices.

In our previous work, we proposed the first transcoding architecture between WZ and H.264 [4] available in the literature. This work improves the H.264 ME using the Motion Vectors (MV) calculated in the Side Information (SI) to reduce the H.264 searching area with negligible RD impact. Nevertheless, in the previous approach we employed a WZ GOP size of 2 to be transcoded to a H.264 GOP size of 2 which means transcoding from K – WZ – K DVC pattern to I – P – I H.264 pattern. This solution is not very useful in a real implementation due to the high bitrate generated because one of every two frames is an I-frame. Moreover the refinement technique is improved for P frames and generalized for longer H.264 patterns. Other improvements of this approach with respect to the previous one is related to the DVC implementation, as, in present work, it is based on VISNET-II project [7] which is more realistic than the architecture used in [4] because it implements lossy key frame coding, on-line correlation noisy modeling and do not use the ideal procedure call at decoder for the stopping criterion. In other words, this work extends the approach presented in [4] to a more realistic GOP size and format implementation. Moreover, this work is a generalization to support whatever GOP size or format incoming from DVC stage to be transcoded to I11P GOP pattern using a low complexity algorithm.

3 Proposed Video Transcoder

In a real scenario, video transcoding should be able to convert efficiently different patterns. For this reason, the main aim of the proposal is to provide an architecture which supports practical GOP patterns making efficiently the transcoding process through exploiting the information that the WZ decoding algorithm provide in order to reduce the H.264 encoding time on the ME process.

Mobile-to-mobile communications need to execute low complexity algorithms at both ends. In the proposed architecture the source employs the DVC encoders and the destination employs the H.264 decoder, so terminal devices support the lower complexity parts in both paradigms (as Figure 1 shows). On the other hand, the

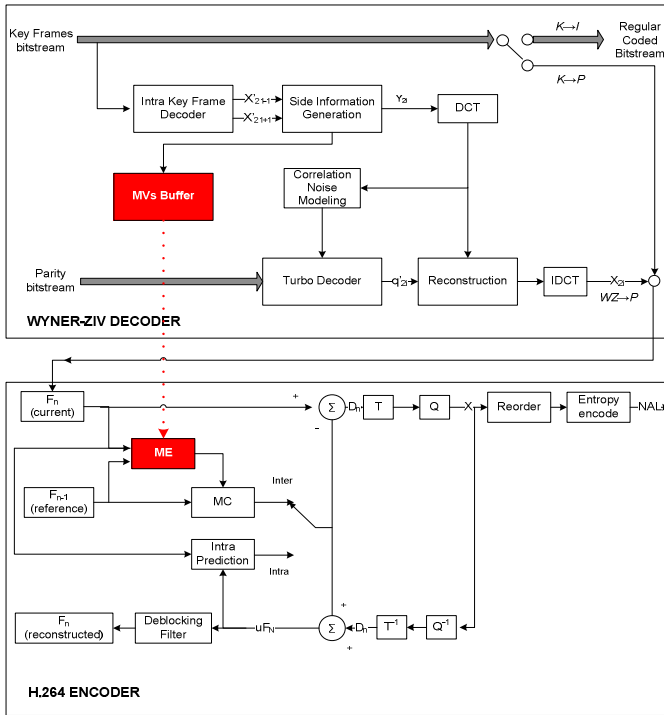


Fig. 2. Proposed WZ to H.264 Video Transcoder

transcoder is allocated in the network where more resources are allowed. The architecture of the proposed transcoder is depicted in Figure 2 and works as follows: the WZ encoder is based on VISNET-II architecture [7] which is an evolution of DISCOVER architecture [8]. In this work we have used Transform Domain (TD). At the encoder, K frames are coded using H.264 Intra and WZ frames are coded following the basis of WZ paradigm [2]. In our experiments we are working with WZ GOP sizes 2, 4 and 8, but the transcoder could accept every GOP length as will be explained in section 3.1. In the second half of the transcoder, there is a H.264 encoder which converts the WZ output into a H.264 bitstream using a GOP I11P because it is the most suitable pattern for mobile-to-mobile video communications. To develop this conversion, every K frame which matches with an I-frame is passed without any transcoding process. On the other hand, every K or WZ frame which matches with a P frame is encoding using the method proposed to reduce the ME time which is explained in section 3.2.

3.1 Mapping GOP Patterns

Side information generation is a crucial task for any WZ codec due to the fact that WZ frames are decoded and reconstructed departing from the SI. There are many

studies about this topic but there are mainly two major approaches: hash-based motion estimation and Motion Compensated Temporal Interpolation (MCTI). In particular, VISNET-II codec employs the latter one. In Figure 3 the first step of SI generation is shown, which consists in matching each forward frame MarcoBlock (MB) with a backward frame MB inside the search area. This matching takes all the possibilities into account and chooses the lowest residual MB. Notice that DVC works with 16x16 partitions to generate the SI (subpartitions are not used) and the search area is defined by a window 32 pixel length. Through this process a MV is obtained for each MB which quantifies the displacement between both MBs, and the middle of this MV represents the displacement for the MB interpolated. The complete SI estimation procedure is detailed in [9].

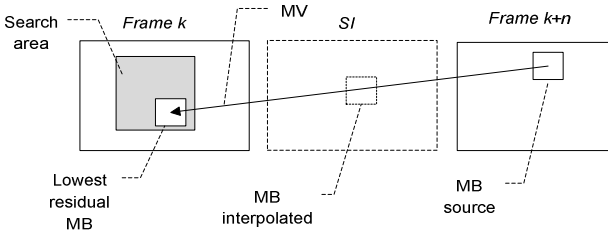


Fig. 3. First step of SI generation process

The present approach proposes to reuse these MVs calculated by WZ algorithm to improve the transcoding process for every WZ GOP to H.264 GOP I11P. Figure 4 represents the transcoding from a WZ GOP 2 to a H.264 GOP I11P. The first K frame is passed to an I-frame without any conversion, as was shown in Figure 2. On the other hand, for every WZ frame a SI is estimated and one MV for each MB. This is shown in the top row where $V_{0,2}$ represents the MVs calculated between K_0 and K_2 to estimate SI_1 for WZ_1 and so on. In other words, $V_{0,2}$ in Figure 4 corresponds with MV in Figure 3. Each MV is divided into two halves and it is applied in H.264 encoding process to accelerate it in the way described in section 3.2. This part is shown in the second row.

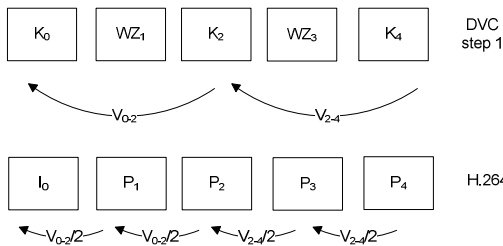


Fig. 4. Mapping from DVC GOP 2 to H.264 GOP I11P

This idea can be extended for longer WZ GOPs but some considerations must be taken into account. Figure 5 shows the transcoding process for a WZ GOP 4. As it is shown in the two top rows, WZ decoding algorithm divides the decoding process into two steps. In the first step, WZ_2 is decoded by calculating the SI between K_0 and K_4 . These MVs (V_{0-4}) are ignored as they have low accuracy. In the second step, there is a reconstruction of WZ_2 (labeled WZ'_2) and now this case is similar to the previous one showed in Figure 4. Then, V_{0-2} and V_{2-4} are divided to improve H.264 encoding procedure. This procedure can be applied for any WZ GOP, including odd GOPs.

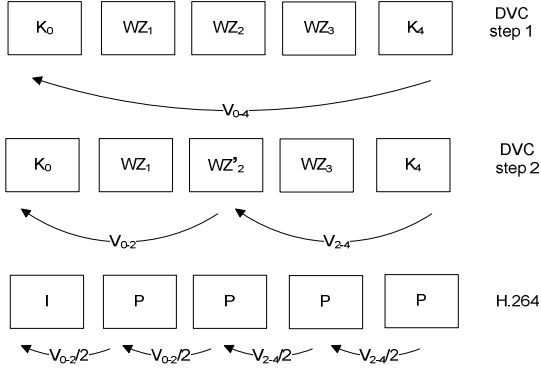


Fig. 5. Mapping from DVC GOP 4 to H.264 GOP I11P

3.2 Motion Estimation Reduction

DVC to H.264 transcoding process joins the largest complexity algorithms of each paradigm, so a lot of effort must be invested in order to improve this task. As it is well known, a big part of this complexity depends largely on the search range used in the H.264 ME process is as a consequence of the quantity of checking done. However, this process may be accelerated because of the search range can be reduced avoiding unnecessary checking without significant impact on quality and bit rate.

To achieve this aim, we propose to reuse the MVs calculated in DVC to define a smaller search range for each MB of H.264 including every sub MB partition. So in this way the checking area is limited by the area S defined in the expression (1).

$$S = \{(x, y) / (x, y) \in (A \cap C)\} \quad (1)$$

where (x, y) are the coordinates to check, A is the search range used by H.264 and C is a circumference which restricts the search with centre on the upper left corner of the MB. C is defined by the equation (2).

$$C^2 = r_x^2 + r_y^2 \quad (2)$$

$$r_x = \max\left(\frac{MV_x}{2}, 2\right) \quad (3)$$

$$r_y = \max\left(\frac{MV_y}{2}, 2\right) \quad (4)$$

Where r_x and r_y are calculated from equations (3) and (4) depending of the MVs halves ($MV_x/2$ and $MV_y/2$) provided by DVC or a minimum value of 2 to avoid applying too small search ranges. Notice that each H.264 subpartition related to a particular H.264 MB takes advance of the same area reduction.

4 Performance Evaluation

The source WZ video 8 was generated by VISNET II codec using a fixed matrix QP = 7 and GOPs 2, 4 and 8. While sequences (in QCIF format) are decoded, the MVs are passed to the H.264 encoder without any increase of complexity. Afterwards, the transcoder converts this WZ video input into a H.264 video stream using QP = 28, 32, 36, 40 as specified in *Bjøntegaard and Sullivan's* common test rule [10][9]. Every WZ GOP pattern was mapped into a H.264 GOP I11P. The simulations were run by using the version JM 14.1 of H.264 and the baseline profile with the default configuration file. The baseline profile was selected because it is the most used profile in real-time applications due to its low complexity. For the same reason, RDOptimization was turned off. For ME, search area was defined by a window with 32 pixel length. In order to check our proposal we have chosen four representative sequences with different motion levels at 15 fps and 30 fps coding 150 frames and 300 respectively, the same sequences that were selected in the DISCOVER codec's evaluation [8]. On the other hand, the *percentage of Time Reduction (%TR)* reported

Table 1. Performance of the proposed transcoder for 15fps sequences

RD performance of the WZ/H.264 video transcoder – 15fps				
Sequence	GOP	Δ PSNR (dB)	Δ Bitrate (%)	TR (%)
Foreman	2	-0.076	2.00	72.75
	4	-0.076	2.04	72.80
	8	-0.073	2.04	73.49
Hall	2	-0.009	0.23	67.03
	4	-0.007	0.16	66.92
	8	-0.006	0.16	66.63
CoastGuard	2	-0.057	1.51	77.97
	4	-0.050	1.35	77.84
	8	-0.055	1.49	78.07
Soccer	2	-0.145	4.11	68.53
	4	-0.150	4.45	69.94
	8	-0.148	4.66	70.11
<i>mean</i>		<i>-0.071</i>	<i>2.02</i>	<i>71.84</i>

displays the average of the times reduction of the four H.264 QP points under study compared to the reference transcoder which is composed by the full WZ decoding and H.264 encoding algorithms.

Table 1 shows the TR of the proposed transcoder for 15fps. TR is over 71% with negligible rate-distortion loss compared to the full complex reference transcoder.

On the other hand, Table 2 includes the results for 30fps sequences. It shows a similar TR (over 72%) improving the RD performance with respect to 15fps sequences.

Although longer DVC GOP patterns may seem to offer a lower performance, it does not happen so and the performance is almost the same. The approach presented here is independent of the DVC GOP size employed. This is because of the way of our algorithm (depicted in section 3) uses the incoming MVs. For longer DVC GOP sizes, the previous MV generated between two K frames are ignored and the algorithm always uses the MV generated in the last step in the WZ decoding process. The side information in this last step is always formed by two frames with distance two between them. These frames can be K frames or frames that have been reconstructed through the entire WZ decoding algorithm which have been reconstructed and improved using the parity bit information sent by the encoder. Therefore, the quality of these frames is better than the original ones which were discarded in the first step of the algorithm.

Table 2. Performance of the proposed transcoder for 15fps sequences

RD performance of the WZ/H.264 video transcoder – 30fps				
Sequence	GOP	Δ PSNR (dB)	Δ Bitrate (%)	TR (%)
Foreman	2	-0.043	1.18	74.29
	4	-0.038	0.97	74.42
	8	-0.044	1.23	74.69
Hall	2	-0.004	0.11	65.84
	4	-0.005	0.09	65.81
	8	-0.004	0.08	65.90
CoastGuard	2	-0.018	0.47	77.35
	4	-0.019	0.49	77.12
	8	-0.022	0.59	76.82
Soccer	2	-0.073	2.21	70.77
	4	-0.082	2.22	71.73
	8	-0.074	2.08	72.05
<i>mean</i>		-0.035	0.98	72.23

In addition, quality values were measured in SSIM terms. SSIM is an improvement of PSNR [11] which is calculated considering luminance similarity, contrast similarity and structural similarity. As it is shown in Figures 6 and 7, using QP = 28, 32, 36 and 40 there are no significant differences of quality and the bit rate obtained by the H.264 reference and our proposed. Similar RD results are obtained comparing with PSNR, as it is shown for 15 fps in Figure 8. As expected, different GOPs do not have an important influence on the quality and the bit rate obtained by both versions. Furthermore, for 30 fps better results are obtained due to the higher frequency.

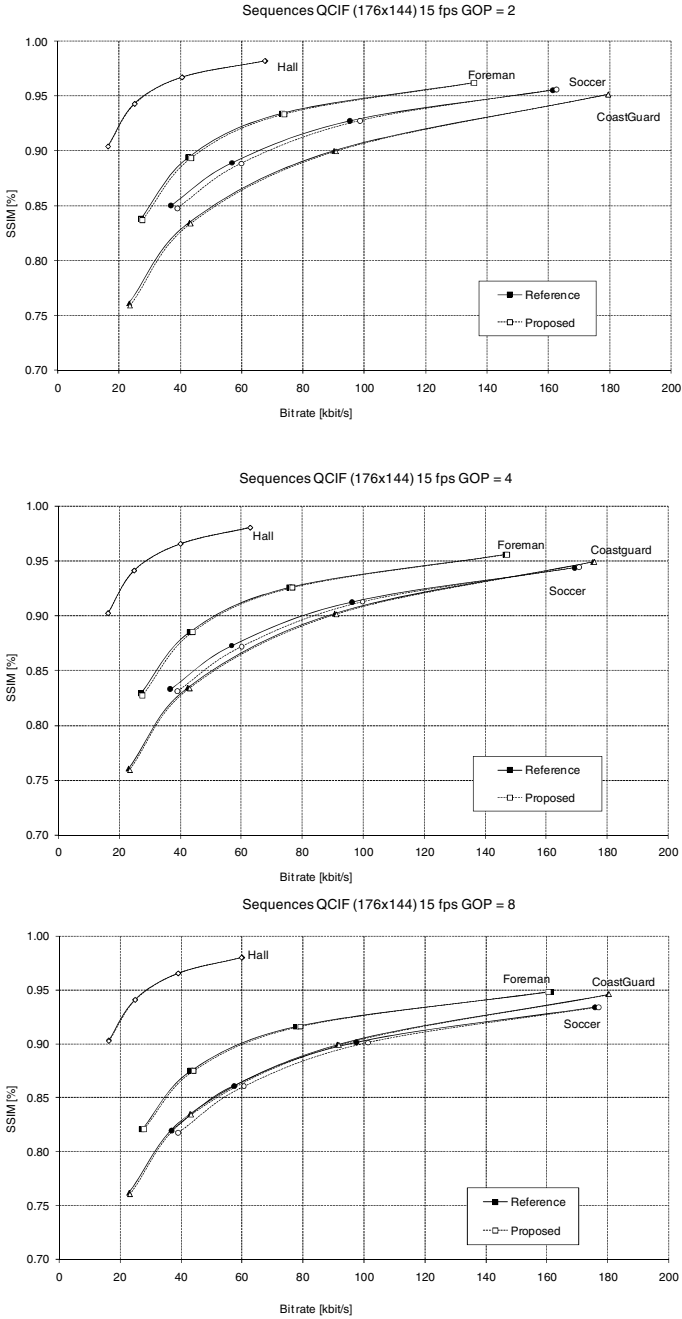


Fig. 6. SSIM/bitrate results using QP=28, 32, 36 and 40 for 15fps sequences with GOP = 2,4,8. Reference symbols: ■Foreman, ◆Hall, ▲ CoastGuard and ●Soccer.

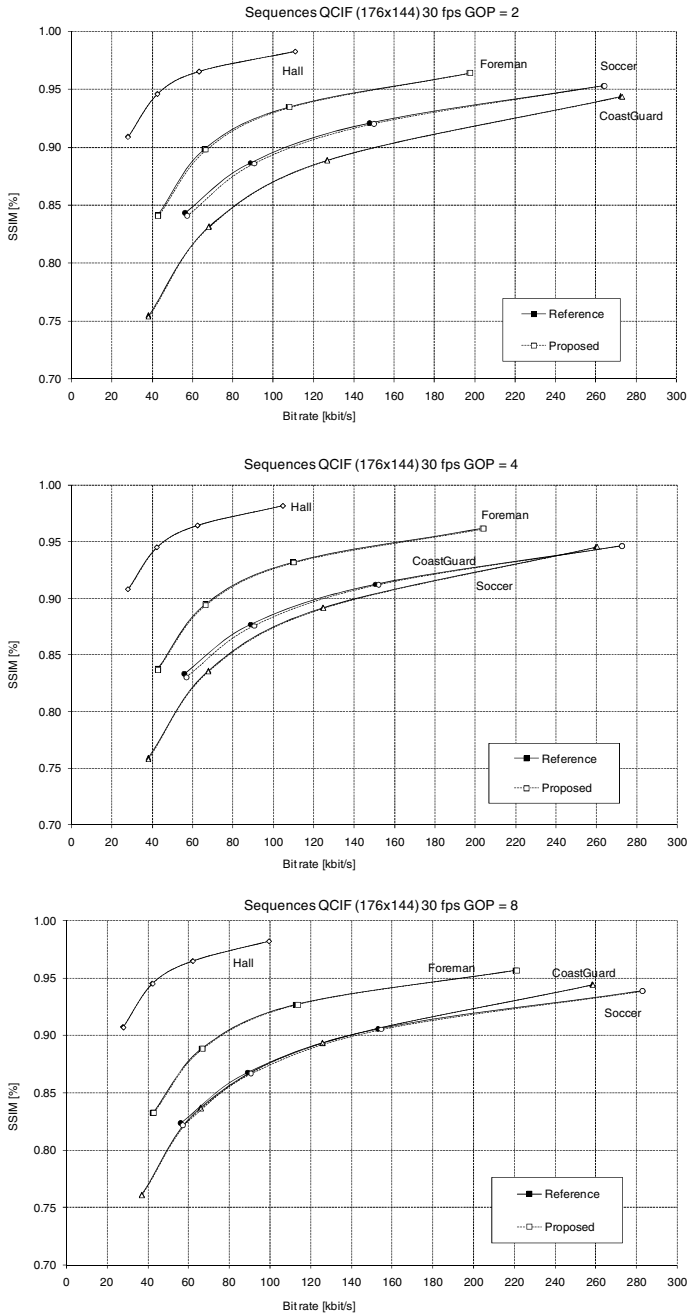


Fig. 7. SSIM/bitrate results using QP=28, 32, 36 and 40 for 30ps sequences with GOP = 2,4,8. Reference symbols: ■Foreman, ◆Hall, ▲CoastGuard and ●Soccer.

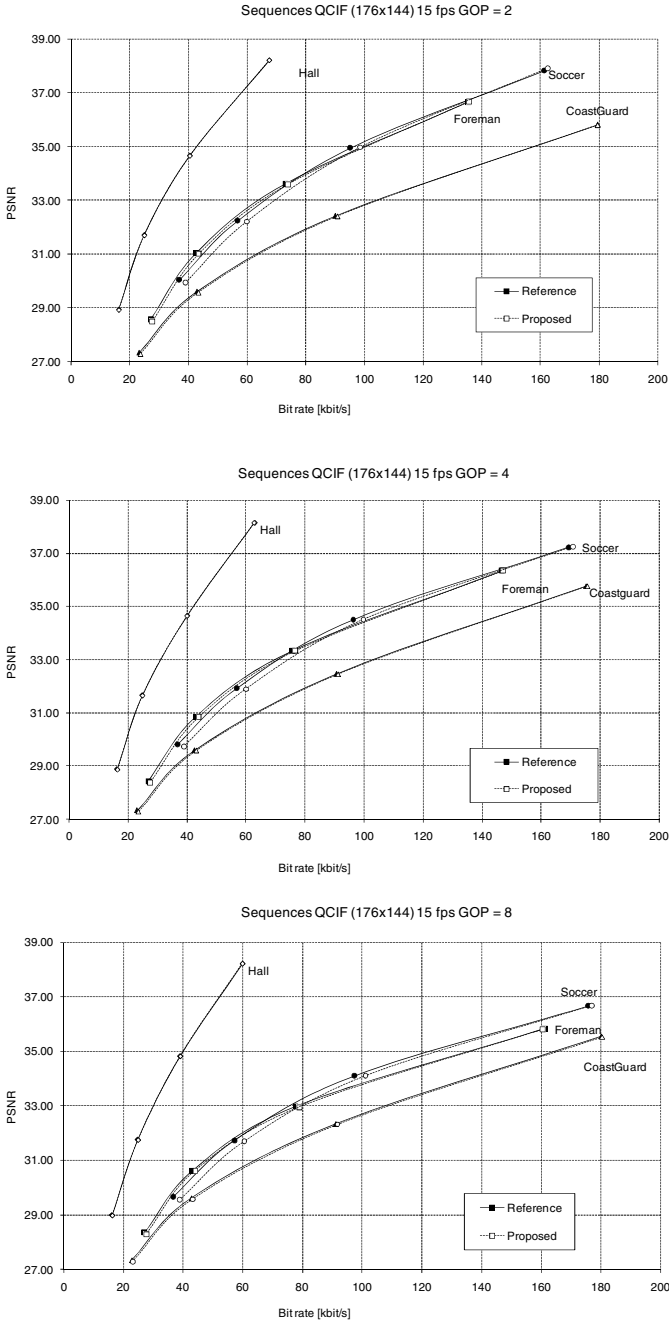


Fig. 8. PSNR/bitrate results using QP=28, 32, 36 and 40 for 15fps sequences with GOP = 2,4,8. Reference symbols: ■Foreman, ◆Hall, ▲CoastGuard and ●Soccer.

5 Conclusions

This work presents a DVC to H.264 transcoder which has been improved to allow every WZ GOP as input and mapping them into a H.264 GOP I11P. Moreover, the transcoding process was improved by reusing the MVs generated at the WZ decoding algorithm to reduce the time spent on the H.264 ME over 72%. As a consequence, the overall time is considerably reduced without any significant loss of RD. As future work, it is planned to extend H.264 pattern introducing B frames. Furthermore, we will invest effort in order to accelerate Wyner-Ziv decoding stage.

Acknowledgments. This work was supported by the Spanish MEC and MICINN, as well as European Commission FEDER funds, under Grants CSD2006-00046, TIN2009-14475-C04. It was also partly supported by JCCM funds under grant PEII09-0037-2328 and PII2109-0045-9916, and the University of Castilla-La Mancha under Project AT20101802. The work presented was developed by using the VISNET2-WZ-IST software developed in the framework of the VISNET II project.

References

1. ITU-T and ISO/IEC JTC 1: Advanced Video Coding for Generic Audiovisual Services. In: ITU-T Rec. H.264/AVC and ISO/IEC 14496-10 Version 8 (2007)
2. Aaron, A., Zhang, R., Girod, B.: Wyner-Ziv Coding for Motion Video. Asilomar Conference on Signals, Systems and Computers, Pacific Grove, USA (2002)
3. Peixoto, E., Queiroz, R.L., Mukherjee, D.: A Wyner-Ziv Video Transcoder. IEEE Trans. Circuits and Systems for Video Technology (to appear in 2010)
4. Martínez, J.L., Kalva, H., Fernández-Escribano, G., Fernando, W.A.C., Cuenca, P.: Wyner-Ziv to H.264 video transcoder. In: 16th IEEE International Conference on Image Processing (ICIP), Cairo, Egypt, pp. 2941–2944 (2009)
5. Wiegand, T., Sullivan, G.J., Bjontegaard, G., Luthra, A.: Overview of the H.264/AVC video coding standard. IEEE Trans. Circuits Syst. Video Technol. 13, 560–576 (2003)
6. Aaron, A., Rane, S., Rebollo-Monedero, D., Girod, B.: Distributed Video Coding. Proceedings of the IEEE 93(1), 71–83 (2005)
7. VISNET II project, <http://www.visnet-noe.org/> (last visited April 2010)
8. Artigas, X., Ascenso, J., Dalai, M., Klomp, S., Kubasov, D., Ouaret, M.: The DISCOVER codec: architecture, techniques and evaluation: In: Picture Coding Symposium, Lisbon, Portugal (2007)
9. Ascenso, J., Brites, C., Pereira, F.: Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding. In: 5th EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services, Smolenice, Slovak Republic (2005)
10. Sullivan, G., Bjontegaard, G.: Recommended Simulation Common Conditions for H.26L Coding Efficiency Experiments on Low-Resolution Progressive-Scan Source Material. ITU-T VCEG, Doc. VCEG-N81(2001)
11. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing 13(4), 600–612 (2004)

Scalable H.264 Wireless Video Transmission over MIMO-OFDM Channels

Manu Bansal¹, Mohammad Jubran², and Lisimachos P. Kondi^{3,*}

¹ Pillsbury Winthrop Shaw Pittman LLP, Intellectual Property Dept., McLean, VA, USA

² Birzeit University, Dept of Electrical Engineering, Birzeit, Palestine

³ University of Ioannina, Dept. of Computer Science, Ioannina, GR-45110, Greece

Abstract. A cross-layer optimization scheme is proposed for scalable video transmission over wireless Multiple Input Multiple Output Orthogonal Frequency Division Multiplexing (MIMO-OFDM) systems. The scalable video coding (SVC) extension of H.264/AVC is used for video source coding. The proposed cross-layer optimization scheme jointly optimizes application layer parameters and physical layer parameters. The objective is to minimize the expected video distortion at the receiver. Two methods have been developed for the estimation of video distortion at the receiver, which is essential for the cross-layer optimization. In addition, two different priority mappings of the SVC scalable layers are considered. Experimental results are provided and conclusions are drawn.

1 Introduction

Recent advances in computer technology, data compression, high-bandwidth storage devices, high-speed networks, and the third and the fourth generation (3G and 4G) wireless technology have made it feasible to provide the delivery of video over multicarrier wireless channels at high data rates [1]. Transmission over Multiple Input Multiple Output (MIMO) channels using Orthogonal Frequency Division Multiplexing (OFDM) provides such high data rates for multimedia delivery and therefore is of great interest in the area of wireless video applications.

Diversity techniques, such as space-time coding (STC) for multiple antenna systems (i.e., MIMO systems) have been proven to help overcome the degradations due to the wireless channels by providing the receiver with multiple replicas of the transmitted signal over different channels. MIMO systems employ orthogonal space-time block codes (O-STBC) [2], [3], which exploit the orthogonality property of the code matrix to achieve the full diversity gain and have the advantage of low complexity maximum-likelihood (ML) decoding.

* This research was supported by a Marie Curie International Reintegration Grant within the 7th European Community Framework Programme.

On the other hand, OFDM mitigates the undesirable effects of a frequency-selective channel by converting it into a parallel collection of frequency-flat subchannels. OFDM is basically a block modulation scheme where a block of N information symbols is transmitted in parallel on N subcarriers. The subcarriers have the minimum frequency separation required to maintain orthogonality of their corresponding time domain waveforms, yet the signal spectra corresponding to the different subcarriers overlap in frequency. Hence, the available bandwidth is used very efficiently. An OFDM modulator can be implemented as an inverse discrete Fourier transform (IDFT) on a block of N information symbols. To mitigate the effects of intersymbol interference (ISI) caused by channel time spread, each block of N IDFT coefficients is typically preceded by a cyclic prefix (CP) or a guard interval consisting of G samples, such that the length of the CP is at least equal to the channel length. As a result, the effects of the ISI are easily and completely eliminated. Recent developments in MIMO techniques promise a significant boost in performance for OFDM systems. A parallel transmission framework for multimedia data over spectrally shaped channels using multicarrier modulation was studied in [4]. A space-time coded OFDM system to transmit layered video signals over wireless channels was presented in [5]. Video transmission with OFDM and the integration of STC with OFDM have been studied recently [6,7,8]. In [9] an optimal resource allocation method was proposed for multilayer wireless video transmission by using the large-system performance analysis results for various multiuser receivers in multipath fading channels. However, the above approaches have not exploited wireless video transmission over MIMO-OFDM systems with bandwidth optimization.

In this paper, we consider the bandwidth constrained transmission of temporal and quality scalable layers of coded video over MIMO-OFDM wireless networks, with optimization of source coding, channel coding and physical layer parameters on a per group of pictures (per-GOP) basis. The bandwidth allocation problem is addressed by minimizing the expected end-to-end distortion (for one GOP at a time) and optimally selecting the quantization parameter (QP), channel coding rate and the constellation for the STBC symbols used in this MIMO-OFDM system. At the source coding side, we use the scalable video coding (SVC) extension of the H.264/AVC standard which has an error-resilient network abstraction layer (NAL) structure and provides superior compression efficiency [10]. The combined scalability provided by the codec is exploited to improve the video transmission over error-prone wireless networks by protecting the different layers with unequal error protection (UEP). In [11,12], we proposed bandwidth optimization algorithms for SVC video transmission over MIMO (non-OFDM) channels using O-STBC.

A good knowledge of the total end-to-end decoder distortion at the encoder is necessary for such optimal allocation. Accordingly, we use the low-delay, low-complexity method for accurate distortion estimation for SVC video as discussed in [11] and also propose a new modified version of this distortion estimation method. The two distortion estimation methods differ in the priority order in which different types of scalability inherent in the SVC codec, namely temporal

and Signal to Noise Ratio (SNR), are considered for estimation purposes. We also propose two different priority mappings of the scalable layers produced by SVC. Comparison results for the two priority mappings are presented for bandwidth constrained and distortion optimized video transmission over a MIMO-OFDM system. The results exemplify the advantages of the use of each priority mapping for different video sequences.

The rest of the paper is organized as follows. In section 2, the proposed system is introduced. In section 3, the scalable extension of H.264 is described. In section 4, the cross-layer optimization problem is formulated and solved. In section 5, the two video distortion estimation methods are discussed. In section 6, the priority mapping of the temporal and FGS layers of SVC is discussed. In section 7, experimental results are presented. Finally, in section 8, conclusions are drawn.

2 System Description

In our packet-based video transmission system, we utilize channel coding followed by orthogonal space-time block coding for MIMO-OFDM systems. After video encoding, the scalable layers of each frame are divided into packets of constant size γ , which are then channel encoded using a 16-bit cyclic redundancy check (CRC) for error detection and rate-compatible punctured convolutional (RCPC) codes for UEP. These channel-encoded packets are modulated with a particular constellation size and further encoded using O-STBC for each subcarrier for transmission over the MIMO wireless system. A 6-ray typical urban (TU) channel model with AWGN is considered (details shown in Table 1) and ML decoding is used to detect the transmitted symbols at each subcarrier, which are then demodulated and channel decoded for error correction and detection. All the error-free packets for each frame are buffered and then fed to the source decoder with error concealment for video reconstruction. For the MIMO-OFDM

Table 1. Six-ray typical urban (TU) channel model

Delay (μ s)	0.0	0.2	0.5	1.6	2.3	5.0
Power (mean)	0.189	0.379	0.239	0.095	0.061	0.037

system used here, we consider $M_t = 2$ transmit and $M_r = 2$ receive antennas. We used the O-STBC design for MIMO-OFDM systems in which two codewords (corresponding to two time instances) are transmitted. The channel is assumed to be quasi-static for these two codeword time periods. The codeword structure is as follows:

$$\mathbf{C}_{OFDM1} = \begin{bmatrix} x_1 & x_2 \\ x_3 & x_4 \\ \vdots & \vdots \\ x_{2N-1} & x_{2N} \end{bmatrix}, \quad (1)$$

and

$$\mathbf{C}_{OFDM2} = \begin{bmatrix} -x_2^* & x_1^* \\ -x_4^* & x_3^* \\ \vdots & \vdots \\ -x_{2N}^* & x_{2N-1}^* \end{bmatrix} \quad (2)$$

where N is the number of subcarriers and $(\cdot)^*$ denotes the complex conjugate. The two codewords take two time instances and each row represents transmission over one subcarrier. Hence, the two codewords together form a 2×2 O-STBC for each subcarrier. In such a design, we gain spatial diversity but no frequency diversity.

The signal model at the j -th receive antenna for the n -th subcarrier at time t ($t = 1, 2$) is given as

$$y_t^j(n) = \sqrt{\frac{\rho}{M_t}} \sum_{i=1}^{M_t} c_t^i(n) h_{ij}(n) + \eta_t^j(n), \quad (3)$$

where ρ is the channel SNR, $c_t^i(n)$ is the energy-normalized transmitted symbol from the i -th transmit antenna at the n -th tone, and $\eta_t^j(n)$ are independent Gaussian random variables with zero mean and variance 1. $h_{ij}(n)$ is the channel frequency response from the i -th transmit antenna to the j -th receive antenna at the n -th tone. t takes values 1 and 2 since there are two codewords that take two time instances, as mentioned earlier. The fading channel is assumed to be quasi-static. We assume that perfect channel state information is known at the receiver, and the ML decoding is used to detect the transmitted symbols independently.

3 Scalable H.264 Codec

In this work, the scalable extension of H.264/AVC is used for video coding. We will use the acronym ‘‘SVC’’ to specifically refer to the scalable extension of H.264/AVC and not to scalable video coding in general. SVC is based on a hierarchical prediction structure in which a GOP consists of a key picture and all other pictures temporally located between the key picture and the previously encoded key picture. These key pictures are considered as the lowest temporal resolution of the video sequence and are called temporal level zero (TL0) and the other pictures encoded in each GOP define different temporal levels (TL1, TL2, and so on). Each of these pictures is represented by a non-scalable base layer (FGS0) and zero or more quality scalable enhancement fine granularity scalability (FGS) layers. The hierarchical coding structure of SVC is shown in Figure [1](#).

4 Optimal Bandwidth Allocation

The bandwidth allocation problem is defined as the minimization of the expected end-to-end distortion by optimally selecting the application layer parameter, QP

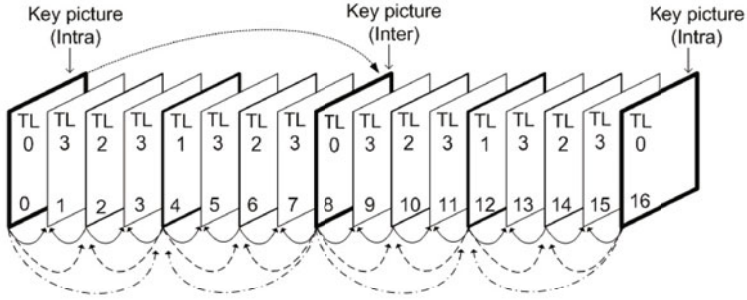


Fig. 1. Hierarchical prediction structure for SVC for a GOP size of 8

value for video encoding, and the physical layer parameters, RCPC coding rate and the symbol constellation choice for the STBC block code. The optimization is considered on a GOP-by-GOP basis and is constrained by the total available bandwidth (symbol rate) B_{budget} . We assume that the SVC codec produces L layers $\mu_1, \mu_2, \dots, \mu_L$ via a combination of temporal and FGS scalability. Then, the bandwidth allocation problem can be described as:

$$\{\mathbf{QP}^*, \mathbf{R}_c^*, \mathbf{M}^*\} = \underset{\{\mathbf{QP}, \mathbf{R}_c, \mathbf{M}\}}{\operatorname{argmin}} E\{D_{s+c}\} \text{ s.t. } B_{s+c} \leq B_{budget} \quad (4)$$

where B_{s+c} is the transmitted symbol rate, B_{budget} is the total available symbol rate and $E\{D_{s+c}\}$ is the total expected end-to-end distortion due to source and channel coding, which needs to be estimated as discussed in Section 5. \mathbf{QP} , \mathbf{R}_c and \mathbf{M} are the admissible set of values for QP, RCPC coding rates and symbol constellations, respectively. For all the layers of each GOP, $\mathbf{QP}^* = \{QP_{\mu_1}, \dots, QP_{\mu_L}\}$, $\mathbf{R}_c^* = \{R_{c,\mu_1}, \dots, R_{c,\mu_L}\}$ and $\mathbf{M}^* = \{M_{\mu_1}, \dots, M_{\mu_L}\}$ define the QP values, the RCPC coding rates and the symbol constellations for each scalable layer, respectively, obtained after optimization. The transmitted symbol rate B_{s+c} can be obtained as

$$B_{s+c} = \sum_{l=1}^L \frac{R_{s,\mu_l}}{R_{c,\mu_l} \times \log_2(M_{\mu_l})} \quad (5)$$

where R_{s,μ_l} is the source coding rate for layer μ_l in bits/s and depends on the quantization parameter value used for that layer; R_{c,μ_l} is the channel coding rate for layer μ_l and is dimensionless; M_{μ_l} is the constellation used by layer μ_l and $\log_2(M_{\mu_l})$ is the number of bits per symbol.

The problem of Eq. (4) is a constrained optimization problem and is solved using the Lagrangian method.

5 Decoder Distortion Estimation

In order to perform the optimization of Eq. (4), it is necessary to estimate the expected video distortion at the receiver $E\{D_{s+c}\}$. In this paper, we use the distortion estimation technique of [11] and we also propose a new technique.

As mentioned previously, SVC produces video frames which are partitioned into FGS layers. We assume that each layer of each frame is packetized into constant size packets of size γ for transmission. At the receiver, any unrecoverable errors in each packet would result in dropping the packet and hence would mean loss of the layer to which the packet belongs. We assume that the channel coding rate and constellation used for the transmission of the base layers of all key pictures is such that they are received error-free. Using the fact that SVC encoding and decoding is done on a GOP basis, it is possible to use the frames within a GOP for error concealment purposes. In the event of losing a frame, temporal error concealment at the decoder is applied such that the lost frame is replaced by the nearest available frame in the decreasing as well as increasing sequential order but from only lower or same temporal levels. We start towards the frames that have a temporal level closer to the temporal level of the lost frame. For the frame in the center of the GOP, the key picture at the start of the GOP is used for concealment.

As discussed in [11], the priority of the base layer (FGS0) of each temporal level decreases from the lowest to the highest temporal level, and each FGS layer for all the frames is considered as a single layer of even lesser priority. We will refer to this method as **Temporal-SNR** scalable decoder distortion estimation (SDDE) method. Alternatively, we can consider both the base and the FGS layers of the reference frames to be used for the encoding and the reconstruction of the frames of higher temporal levels (non-key pictures). In such a case, both the base and the FGS layers of the reference frames (from the lower temporal levels) are considered of the same importance, and of higher importance than the frame(s) (from a higher temporal level) to be motion-compensated and reconstructed. We will refer to this case as the **SNR-Temporal** SDDE method. Next we will present the derivations of the two above-mentioned SDDE methods.

5.1 Temporal-SNR SDDE

In the following derivation of the Temporal-SNR SDDE method, we consider a base layer and one FGS layer. We assume that the frames are converted into vectors via lexicographic ordering and the distortion of each macroblock (and hence, each frame) is the summation of the distortion estimated for all the pixels in the macroblock of that frame. Let f_n^i denote the original value of pixel i in frame n and \hat{f}_n^i denote its encoder reconstruction. The reconstructed pixel value at the decoder is denoted by \tilde{f}_n^i . The mean square error for this pixel is defined as [13]:

$$d_n^i = \mathbb{E} \left\{ \left(f_n^i - \tilde{f}_n^i \right)^2 \right\} = \left(f_n^i \right)^2 - 2f_n^i \mathbb{E} \left\{ \tilde{f}_n^i \right\} + \mathbb{E} \left\{ \left(\tilde{f}_n^i \right)^2 \right\} \quad (6)$$

where d_n^i is the distortion per pixel. The base layers of all the key pictures are assumed to be received error-free. The s^{th} moment of the i^{th} pixel of the key pictures n is calculated as

$$\mathbb{E} \left\{ \left(\tilde{f}_n^i \right)^s \right\} = P_{nE1} \left(\hat{f}_{nB}^i \right)^s + (1 - P_{nE1}) \left(\hat{f}_{n(B,E1)}^i \right)^s \quad (7)$$

where \hat{f}_{nB}^i , $\hat{f}_{n(B,E1)}^i$ are the reconstructed pixel values at the encoder using only the base layer, and the base along with the FGS layer of frame n , respectively. P_{nE1} is the probability of losing the FGS layer of frame n .

For all the frames except the key pictures of a GOP, let us denote $\hat{f}_{nB-u_n v_n}^i$ as the i^{th} pixel value of the base layer of frame n reconstructed at the encoder. Frames $u_n (< n)$ and $v_n (> n)$ are the reference pictures used in the hierarchical prediction structure for the reconstruction of frame n . We will refer to these frames (u_n and v_n) as the ‘‘true’’ reference pictures for frame n . In the decoding process of SVC, the frames of each GOP are decoded in the order starting from the lowest to the highest temporal level. At the decoder, if either or both of the true reference frames are not received correctly, the non-key picture(s) will be considered erased and will be concealed.

For the Temporal-SNR SDDE method, the s^{th} moment of the i^{th} pixel of frame n when at least the base layer is received correctly is defined as

$$E \left\{ \left(\tilde{f}_n^i(u_n, v_n) \right)^s \right\} = (1 - P_{u_n})(1 - P_{v_n}) P_{nE1} \left(\hat{f}_{nB-u_n v_n}^i \right)^s + (1 - P_{u_n})(1 - P_{v_n})(1 - P_{nE1}) \left(\hat{f}_{n(B,E1)-u_n v_n}^i \right)^s \quad (8)$$

where, P_{u_n} and P_{v_n} are the probabilities of losing the base layer of the reference frames u_n and v_n , respectively. Now to get the distortion per-pixel after error concealment, we define a set $\mathbf{Q} = \{f_n, f_{q1}, f_{q2}, f_{q3}, \dots, f_{GOPend}\}$, where f_n is the frame to be concealed, f_{q1} is the first frame, f_{q2} is the second frame to be used for concealment of f_n , and so on till one of the GOP ends is reached. The s^{th} moment of the i^{th} pixel using the set \mathbf{Q} is defined as

$$E \left\{ \left(\tilde{f}_n^i \right)^s \right\} = (1 - P_n) E \left\{ \left(\tilde{f}_n^i(u_n, v_n) \right)^s \right\} + (1 - \bar{P}_n)(1 - P_{q1}) E \left\{ \left(\tilde{f}_{q1}^i(u_{q1}, v_{q1}) \right)^s \right\} + (1 - \bar{P}_n \bar{P}_{q1})(1 - P_{q2}) E \left\{ \left(\tilde{f}_{q2}^i(u_{q2}, v_{q2}) \right)^s \right\} + \dots + \left(1 - \bar{P}_n \prod_{z=1}^{|\mathbf{Q}|-2} \bar{P}_{qz} \right) E \left\{ \left(\tilde{f}_{GOPend}^i \right)^s \right\} \quad (9)$$

where $\bar{P}_n = (1 - P_n)(1 - P_{u_n})(1 - P_{v_n})$ is the probability of correctly receiving the base layers of frame n and the base layers of its reference pictures.

5.2 SNR-Temporal SDDE

Similar to the Temporal-SNR SDDE case, in this method the base layer of all the key pictures are assumed to be received error-free and the s^{th} moment of the i^{th} pixel of the key pictures n is again calculated using Eq. (7). For all the frames except the key pictures of a GOP, let us denote $\hat{f}_{nB-u_{(B,E1)n} v_{(B,E1)n}}^i$ as the i^{th} pixel value of the base layer of frame n reconstructed at the encoder. Frames $u_{(B,E1)n} (< n)$ and $v_{(B,E1)n} (> n)$ are the reference pictures (including both base and FGS layers) used in the hierarchical prediction structure for the

reconstruction of frame n . In case of losing the FGS layers of the reference pictures, only the base layers of the frames u_n and v_n are used as the reference for frame n . As discussed above, in SVC the decoding of all the frames in a GOP is done from the lowest to the highest temporal level. Similar to the Temporal-SNR method, we will use the “true” reference frames for distortion estimation and hence, the loss of base layer of either or both the reference frames will result in the concealment of the frame n . The s^{th} moment of the i^{th} pixel of frame n when at least the base layer is received correctly is calculated as:

$$\begin{aligned}
E \left\{ \left(\tilde{f}_n^i(u_n, v_n) \right)^s \right\} &= P_{uvB} P_{nE1} \left(\hat{f}_{nB-u_{Bn}v_{Bn}}^i \right)^s \\
&+ P_{uvB} (1 - P_{nE1}) \left(\hat{f}_{n(B,E1)-u_{Bn}v_{Bn}}^i \right)^s \\
&+ P_{uvB,E1} P_{nE1} \left(\hat{f}_{nB-u_{(B,E1)n}v_{(B,E1)n}}^i \right)^s \\
&+ P_{uvB,E1} (1 - P_{nE1}) \left(\hat{f}_{n(B,E1)-u_{(B,E1)n}v_{(B,E1)n}}^i \right)^s
\end{aligned} \tag{10}$$

where, $P_{uvB} = (1 - P_{u_nB})(1 - P_{v_nB})P_{u_nE1}P_{v_nE1}$ is the probability of correctly receiving the base layers and not receiving the FGS layers of the frames u_n and v_n . Similarly, $P_{uvB,E1} = (1 - P_{u_nB})(1 - P_{v_nB})(1 - P_{u_nE1})(1 - P_{v_nE1})$ is the probability of correctly receiving the base layers and the FGS layers of the frames u_n and v_n . In case the base layer of frame n is lost, the complete frame has to be concealed. To get the distortion per-pixel after error concealment, we use Eq. (9).

The performance of the two SDDE methods is evaluated by comparing it with the actual decoder distortion estimation averaged over 200 channel realizations. Different video sequences encoded at 30 fps, GOP size of eight frames and six layers are used in packet-based video transmission simulations. Each of these layers is considered to be affected with different loss rates $P = \{P_{TL0}, P_{TL1}, P_{TL2}, P_{TL3}, P_{E1}\}$, where P_{TLx} is the probability of losing the base layer of a frame that belongs to TLx and P_{E1} is the probability of losing FGS1 of a frame. For performance evaluation, packet loss rates considered are $P1 = \{0\%, 0\%, 5\%, 5\%, 10\%\}$ and $P2 = \{0\%, 10\%, 20\%, 30\%, 50\%\}$. In Table 1, the average Peak Signal to Noise Ratio (PSNR) performance is presented for the “Foreman”, “Akiyo” and “Carphone” sequences. As can be observed, both the Temporal-SNR and the SNR-Temporal methods result in good average PSNR estimates and hence they are used to solve the optimization problem of section 4.

Table 2. Average PSNR comparison for the proposed distortion estimation algorithms

	Foreman 363 kbps	Akiyo 268 kbps	Carphone 612 kbps
Actual P1 (dB)	36.40	45.91	40.85
Temporal-SNR SDDE (dB)	35.48	45.84	40.12
SNR-Temporal SDDE (dB)	36.00	45.43	40.35
Actual P2 (dB)	30.82	41.46	35.32
Temporal-SNR SDDE (dB)	29.80	41.20	35.10
SNR-Temporal SDDE (dB)	30.22	40.86	35.28

6 Priority Mapping of Scalable Layers

We considered two different mappings of the temporal and FGS layers of SVC into the scalable layers μ_i . Let us assume that the number of temporal layers is T . For a GOP size of 8, as used here, we have $T = 4$. For the first mapping, which we call the Temporal-SNR mapping, the first $L - 1$ layers (μ_1, \dots, μ_{L-1}) are the base layers (FGS0) of the frames associated with the lowest to the highest temporal level in decreasing order of importance for video reconstruction. So, $L - 1 = T$ and the number of scalable layers is $L = T + 1$. The FGS layer of all the frames in a GOP are defined as a single layer μ_L of even lesser importance. The Temporal-SNR distortion estimation method uses exactly the same priorities as the Temporal-SNR mapping, so we used it for our experimental results for this mapping. The second mapping is the SNR-temporal mapping. For this mapping, there are two layers, base and enhancement for each of the T temporal layers. In this case, the FGS layer of the lower temporal layer has more importance than the base layer of the higher temporal levels. Thus, there are a total of $L = 2T$ scalable layers. For the SNR-Temporal mapping, we used the SNR-Temporal distortion estimation method, as it uses exactly the same priorities.

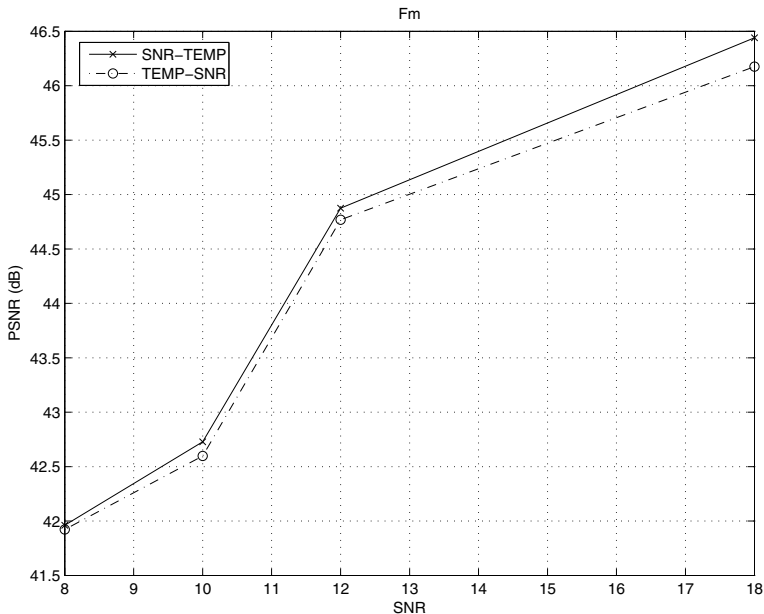


Fig. 2. Performance of the cross-layer optimization using the Temporal-SNR and the SNR-Temporal mappings of scalable layers (“Foreman” sequence)

7 Experimental Results

For experimental results, the “Foreman” and ‘Akiyo’ video sequences are encoded at 30 fps, GOP=8 and constant Intra-update (I) at every 32 frames. We

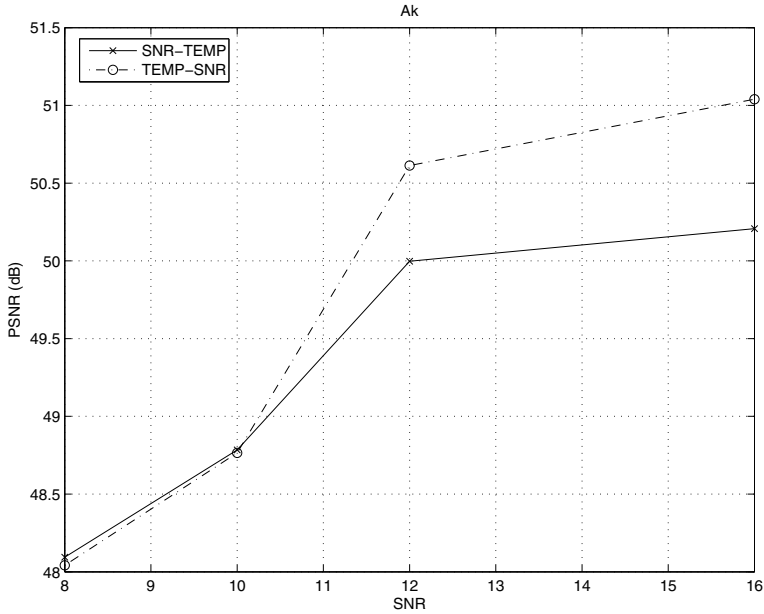


Fig. 3. Performance of the cross-layer optimization using the Temporal-SNR and the SNR-Temporal mappings of scalable layers (“Akiyo” sequence)

consider the video encoding QP values in the range of 16 to 50 and RCPC coding rates of $\mathbf{R}_c = 8/K : K \in \{32, 28, 24, 20, 16, 12\}$, which are obtained by puncturing a mother code of rate $8/32$ with constraint length of 3 and a code generator $[23;35;27;33]_o$. Quadrature amplitude modulation (QAM) is used with the possible constellations size $\mathbf{M}=\{4, 8, 16\}$. The total number of subcarriers N for the OFDM system is fixed at 64. This includes a cyclic prefix (CP) of $1/8$ and guard interval (GI) of $1/8$ of the total number of subcarriers. The packet size is chosen as $\gamma = 100$ bytes. Both the Temporal-SNR and SNR-Temporal priority mappings are considered.

Average PSNR results obtained for transmission of the “Foreman” sequence over the MIMO-OFDM system after the optimal selection of the application layer and physical layer parameters (on a GOP-by-GOP basis) for a channel SNR of 8dB, 10dB, 12 dB and 18dB are shown in Figure 2. Overall, we can see that the SNR-Temporal mapping performs better (in the PSNR sense) than the Temporal-SNR mapping.

Similarly, in Figure 3, we show the average PSNR value comparison of the SNR-Temporal and Temporal-SNR mappings for the “Akiyo” sequence. The PSNR results are obtained after the optimal parameter selection for a channel SNR of 8dB, 10db, 12dB and 16dB. However, we can clearly see that the behavior (in the PSNR sense) for a low motion sequence is opposite compared to the previous case, i.e., the Temporal-SNR mapping performs better than the SNR-Temporal mapping.

8 Conclusions

We have proposed a novel cross-layer optimization scheme for wireless video transmission over MIMO-OFDM channels. The scalable extension of H.264 is used for source coding and the compressed video is divided into scalable layers. For each of these scalable layers, the cross-layer optimization scheme determines the quantization parameter, channel coding rate, and symbol constellation. In order to carry out the optimization, an accurate estimation of the expected video distortion at the receiver is required. We have developed two expected distortion estimation methods, the Temporal-SNR SDDE method and the SNR-Temporal SDDE method. These methods differ in the priority order in which temporal and SNR scalability are considered. We have also proposed two different priority mappings of the scalable layers, the Temporal-SNR mapping and the SNR-Temporal mapping. We have presented experimental results that show the outcome of the cross-layer optimization using both distortion estimation methods. The SNR-Temporal mapping performs better for high-motion video sequences, while the Temporal-SNR mapping performs better for low-motion video sequences.

References

1. Wang, H., Kondi, L.P., Luthra, A., Ci, S.: 4G Wireless Video Communications. John Wiley and Sons, Ltd., Chichester (2009)
2. Alamouti, S.M.: A simple transmit diversity technique for wireless communications. *IEEE Journal on Selected Areas in Communications* 16(8), 1451–1458 (1998)
3. Tarokh, V., Seshadri, N., Calderbank, A.R.: Space-time block codes from orthogonal designs. *IEEE Transactions on Information Theory* 45(5), 1456–1467 (1999)
4. Zheng, H., Liu, K.J.R.: Robust image and video transmission over spectrally shaped channels using multicarrier modulation. *IEEE Transactions on Multimedia* (March 1999)
5. Kuo, C., Kim, C., Kuo, C.C.J.: Robust video transmission over wideband wireless channel using space-time coded OFDM systems. In: *Proc. WCNC*, vol. 2 (March 2002)
6. Zhang, H., Xia, X.-G., Zhang, Q., Zhu, W.: Precoded OFDM with adaptive vector channel allocation for scalable video transmission over frequency-selective fading channels. *IEEE Trans. Mobile Computing* 1(2) (June 2002)
7. Kuo, C., Kim, C., Kuo, C.-C.J.: Robust video transmission over wideband wireless channel using space-time coded OFDM systems. In: *Proc. IEEE Wireless Comm. and Networking Conf.*, WCNC 2002 (March 2002)
8. Dardari, D., Martini, M.G., Milantoni, M., Chiani, M.: MPEG-4 video transmission in the 5Ghz band through an adaptive ofdm wireless scheme. In: *Proc. 13th IEEE Intl Symp. Personal Indoor, and Mobile Radio Comm.*, vol. 4 (2002)
9. Zhao, S., Xiong, Z., Wang, X.: Optimal resource allocation for wireless video over CDMA networks. *IEEE Trans. Mobile Computing* 4(1) (January 2005)
10. Schwarz, H., Marpe, D., Wiegand, T.: Overview of the scalable video coding extension of the H.264/AVC standard. *IEEE Transactions on Circuits and Systems for Video Technology* 17(9), 1103–1120 (2007)

11. Jubran, M.K., Bansal, M., Kondi, L.P.: Low-delay low-complexity bandwidth-constrained wireless video transmission using SVC over MIMO systems. *IEEE Transactions on Multimedia* 10(8), 1698–1707 (2008)
12. Jubran, M.K., Bansal, M., Kondi, L.P., Grover, R.: Accurate distortion estimation and optimal bandwidth allocation for scalable H.264 video transmission over MIMO systems. *IEEE Transactions on Image Processing* 18(1), 106–116 (2009)
13. Zhang, R., Regunathan, S.L., Rose, K.: Video coding with optimal inter/intra-mode switching for packet loss resilience. *IEEE Journal on Selected Areas in Communications* 18(6), 966–976 (2000)

A GPU-Accelerated Real-Time NLMeans Algorithm for Denoising Color Video Sequences

Bart Goossens, Hiệp Luong, Jan Aelterman, Aleksandra Pižurica,
and Wilfried Philips*

Ghent University, TELIN-IPI-IBBT
St.-Pietersnieuwstraat 41, 9000 Ghent, Belgium

Abstract. The NLMeans filter, originally proposed by Buades et al., is a very popular filter for the removal of white Gaussian noise, due to its simplicity and excellent performance. The strength of this filter lies in exploiting the repetitive character of structures in images. However, to fully take advantage of the repetitivity a computationally extensive search for similar candidate blocks is indispensable. In previous work, we presented a number of algorithmic acceleration techniques for the NLMeans filter for still grayscale images. In this paper, we go one step further and incorporate both temporal information and color information into the NLMeans algorithm, in order to restore video sequences. Starting from our algorithmic acceleration techniques, we investigate how the NLMeans algorithm can be easily mapped onto recent parallel computing architectures. In particular, we consider the graphical processing unit (GPU), which is available on most recent computers. Our developments lead to a high-quality denoising filter that can process DVD-resolution video sequences in real-time on a mid-range GPU.

1 Introduction

Noise in digital video sequences generally originates from the analogue circuitry (e.g. camera sensors and amplifiers) in video cameras. The noise is mostly visible in bad lighting conditions and using short camera sensor exposure times. Also, video sequences transmitted over analogue channels or stored on magnetic tapes, are often subject to a substantial amount of noise. In the light of the large scale digitization of analogue video material, noise suppression becomes desirable, both to enhance video quality and compression performance.

In the past decades, several denoising methods have been proposed for noise removal, for still images (e.g. [1, 2, 11, 3, 4, 5, 6, 11]) or particularly for video sequences (see [7, 8, 9, 10, 11, 12, 13, 14]). Roughly speaking, these video denoising methods can be categorized into:

1. *Spatially and temporally local methods* (e.g. [8, 11]): these methods only exploit image correlations in local spatial and temporal windows of fixed size

* B. Goossens and A. Pižurica are postdoctoral researchers of the Fund for Scientific Research in Flanders (FWO), Belgium.

(based on sparsity in a multiresolution representation). The temporal filtering can either be causal or non-causal. In the former case, only past frames are used for filtering. In the latter case, future frames are needed, which can be achieved by introducing a temporal delay¹.

2. *Spatially local methods with recursive temporal filtering* [9, 10, 14, 15]: these methods rely on recursive filtering that takes advantage of the temporal correlations between subsequent frames. Because usually, first order (causal) infinite impulse response filters are used and no temporal delay is required.
3. *Spatially and temporally non-local methods* [12, 13]: these methods take advantage of repetitive structures that occur both spatially and temporally. Because of computation time and memory restrictions, in practice these methods make use of a *search window* (this is a spatio-temporal window in which similar patches are being searched for). By the practical restrictions, the methods actually fall under the first class, however we expect that by more efficient parallel computing architectures and larger RAM memory the non-locality of these methods will further be extended in the future.

One popular filter that makes use of the repetitive character of structures in video sequences and hence belongs to the third class, is the NLMMeans filter [16]. Suppose that an unknown video signal $\mathbf{X}(\mathbf{p})$ is corrupted by an additive noise process $\mathbf{V}(\mathbf{p})$, resulting in the observed video signal:

$$\mathbf{Y}(\mathbf{p}) = \mathbf{X}(\mathbf{p}) + \mathbf{V}(\mathbf{p}) \quad (1)$$

Here, $\mathbf{p} = [p_x, p_y, p_t]$ is the spatio-temporal position within the video sequence. $\mathbf{X}(\mathbf{p})$, $\mathbf{Y}(\mathbf{p})$ and $\mathbf{V}(\mathbf{p})$ are functions that map values from \mathbb{Z}^3 onto the RGB color space \mathbb{R}^3 . The NLMMeans video filter estimates the denoised value of $\mathbf{X}(\mathbf{p})$ as the weighted average of all pixel intensities in the video sequence:

$$\hat{\mathbf{X}}(\mathbf{p}) = \frac{\sum_{\mathbf{q} \in \delta} w(\mathbf{p}, \mathbf{p} + \mathbf{q}) \mathbf{Y}(\mathbf{p} + \mathbf{q})}{\sum_{\mathbf{q} \in \delta} w(\mathbf{p}, \mathbf{p} + \mathbf{q})}, \quad (2)$$

where $\mathbf{q} = [q_x, q_y, q_t]$ and where the weights $w(\mathbf{p}, \mathbf{p} + \mathbf{q})$ depend on the similarity of patches centered at positions \mathbf{p} and $\mathbf{p} + \mathbf{q}$. δ is a three dimensional search window in which similar patches are searched for. For simplicity of the notation, we assume that $\mathbf{Y}(\mathbf{p} + \mathbf{q})$ is everywhere defined in (2). In practice, we make use of boundary extension techniques (e.g. mirroring) near the image boundaries. Because of the high computational complexity of the NLMMeans algorithm (the complexity is quadratic in the number of pixels in the video sequence and linear in the patch size) and because of the fact that the original NLMMeans method performed somewhat inferior compared to other (local) state-of-the-art denoising method, many improvements have been proposed by different researchers. Some of these improvements are better similarity measures [17, 18, 19], adaptive patch sizes [20], and algorithmic acceleration techniques [4, 19, 21, 22].

¹ A temporal delay is not desirable for certain applications, such as video communication.

In our previous work [4], we proposed a number of improvements to the NLMeans filter, for denoising grayscale still images. Some of these improvements which are relevant for this paper are:

- An extension of the NLMeans to correlated noise: even though the original NLMeans filter relies on a *white* Gaussian noise assumption, the power spectral densities of noise in *real* images and video sequences is rarely flat (see [23]).
- Acceleration techniques that exploit the symmetry in the weight computation and that compute the Euclidean distance between patches by a recursive moving average filter. By these accelerations, the computation time can be reduced by a factor 121 (for 11×11 patches), without sacrificing image quality at all!

In spite of efforts by many researchers and also our recent improvements, the NLMeans algorithm is not well suited for real-time denoising of video sequences on a CPU. Using our improvements, denoising one 512×512 color image takes about 30 sec. for a modestly optimized C++ implementation on a recent 2GHz CPU (single-threaded implementation). Consequently this technique is not applicable to e.g. real-time video communication.

Nowadays, there is a trend toward the use of parallel processing architectures in order to accelerate the processing. One example of such architecture is the graphical processing unit (GPU). Although the GPU is primarily designed for the rendering of 3D scenes, advances of the GPU in the late 90's enabled many researchers and engineers to use the GPU for more general computations. This led to the so-called GPGPU (General-Purpose computations on GPUs) [24] and many approaches (e.g. based on OpenGL, DirectX, CUDA, OpenCL, ...) exist to achieve GPGPU with existing GPU hardware. Also because the processing power of modern GPUs has tremendously increased in the last decade (even for inexpensive GPUs a speed-up of a factor $20 \times$ to $100 \times$ can be expected) and is even more improving, it becomes worthwhile to investigate which video denoising methods can efficiently be implemented on a GPU.

Recently, a number of authors have implemented the NLMeans algorithm on a GPU: in [25] a locally constant weight assumption is used in the GPU implementation to speed up the basic algorithm. In [26], a GPU extension of the NLMeans algorithm is proposed to denoise ultrasound images. In this approach, the maximum patch size is limited by the amount of shared memory of the GPU.

In this paper, we focus on algorithmic acceleration techniques for the GPU without sacrificing denoising quality, i.e., the GPU implementation computes the exact NLMeans formula, and without patch size restrictions imposed by the hardware. To do so, we first review how NLMeans-based algorithms can be mapped onto parallel processing architectures. We will find that the core ideas of our NLMeans algorithmic acceleration techniques are directly applicable, but the algorithms themselves need to be modified. By these modifications, we will see that the resulting implementation can process DVD video in real-time on a mid-range GPU. Next, as a second contribution of this paper, we explain how

the filter can be used to remove correlated noise (both spatially as across color channels) from video sequences.

The outline of this paper is as follows: on Section 2, we first review some basic GPGPU programming concepts. Next, we develop an efficient NLMeans algorithm for a GPU and its extension to deal with noise which is correlated across color channels. In Section 3 we give experimental results our method. Finally, Section 4 concludes this paper.

2 An Efficient NLMeans Algorithm for a GPU

In this Section, we will explain the algorithmic improvements that we made to the NLMeans filter in order to efficiently run the algorithm on a GPU. As already mentioned, many approaches and/or programming language extensions exist for GPGPU programming. Because the GPU technology is quickly evolving, we will present a description of the algorithm that is quite general and that does not rely on specific GPU technology choices. This way, the algorithms we present can still be useful in the future, when newer GPU architectures become available.

2.1 General GPGPU Concepts

One core element in GPGPU techniques is the so-called *kernel function*. A *kernel function* is a function that evaluates the output pixel intensities for a specific position in the output image (or even multiple output images) and that takes as input both the position (\mathbf{p}) in the video sequence, and a number of input images (which we will denote as $\mathbf{U}_1^{(i)}, \dots, \mathbf{U}_K^{(i)}$). A GPGPU program can then be considered to be a cascade of kernel functions $\mathbf{f}^{(I)} \circ \mathbf{f}^{(I-1)} \circ \dots \circ \mathbf{f}^{(1)}$ applied to a set of input images. Mathematically, the evaluation of one such kernel function (which we will call a *pass*) can be expressed as:

$$\left[\mathbf{U}_1^{(i+1)}, \dots, \mathbf{U}_K^{(i+1)} \right] (\mathbf{p}) = \mathbf{f}_{\mathbf{U}_1^{(i)}, \dots, \mathbf{U}_K^{(i)}}^{(i)} (\mathbf{p}) \quad (3)$$

where the kernel function takes as input the output images of the previous pass, $\mathbf{U}_1^{(i)}, \dots, \mathbf{U}_K^{(i)}$ and subsequently computes the inputs for the next pass, $\mathbf{U}_1^{(i+1)}, \dots, \mathbf{U}_K^{(i+1)}$. More specifically, the kernel function $\mathbf{f}^{(i)}$ maps a spatio-temporal coordinate (\mathbf{p}) onto a three-dimensional RGB color vector.

Now, porting an algorithm to the GPU comes down to converting the algorithm into a finite, preferably low number of passes as defined in (3) and with fairly simple functions $\mathbf{f}^{(i)}$:

$$\begin{aligned} \left[\mathbf{U}_1^{(2)}, \dots, \mathbf{U}_K^{(2)} \right] (\mathbf{p}) &= \mathbf{f}_{\mathbf{U}_1^{(1)}, \dots, \mathbf{U}_K^{(1)}}^{(1)} (\mathbf{p}), \\ \left[\mathbf{U}_1^{(3)}, \dots, \mathbf{U}_K^{(3)} \right] (\mathbf{p}) &= \mathbf{f}_{\mathbf{U}_1^{(2)}, \dots, \mathbf{U}_K^{(2)}}^{(2)} (\mathbf{p}), \\ &\vdots \\ \left[\mathbf{U}_1^{(I+1)}, \dots, \mathbf{U}_K^{(I+1)} \right] (\mathbf{p}) &= \mathbf{f}_{\mathbf{U}_1^{(I)}, \dots, \mathbf{U}_K^{(I)}}^{(I)} (\mathbf{p}). \end{aligned} \quad (4)$$

We remark that not all passes need to process all input images, i.e. it is completely legal that $\mathbf{U}_1^{(i+1)} = \mathbf{U}_1^{(i)}$. In this case, we express this formally by saying that the function $\mathbf{f}_{\mathbf{U}_1^{(i)}, \dots, \mathbf{U}_K^{(i)}}^{(i)}(\mathbf{p})$ is constant in $\mathbf{U}_1^{(i)}$.

2.2 Straightforward GPU Implementation of the NLMeans Filter

First, we will show that a straightforward (naive) implementation of the traditional NLMeans filter from [13, 16] leads to a very high number of passes, hence an algorithm that is inefficient even on the GPU. Next, we will explain how our own algorithmic accelerations can be converted into a program for the GPU as in equation (4). We will do this for a broad range of weighting functions that are a function of the Euclidean distance measure between two patches:

$$w(\mathbf{p}, \mathbf{p} + \mathbf{q}) = g \left(\sum_{(\Delta x, \Delta y) \in [-B, \dots, B]^2} \left\| \mathbf{r}_{\mathbf{p}, \mathbf{q}}^{(\Delta x, \Delta y)} \right\|^2 \right) \quad (5)$$

with $\mathbf{r}_{\mathbf{p}, \mathbf{q}}^{(\Delta x, \Delta y)} = \mathbf{Y}(p_x + q_x + \Delta x, p_y + q_y + \Delta y, p_t + q_t) - \mathbf{Y}(p_x + \Delta x, p_y + \Delta y, p_t)$, with $(2B + 1) \times (2B + 1)$ the patch size and where the function $g(r)$ has the property that $g(0) = 1$ (such that the weight $w = 1$ if the Euclidean distance between two patches is zero, i.e., for similar patches) and $\lim_{r \rightarrow \infty} g(r) = 0$ (the weight $w = 0$ for dissimilar patches). In particular, we consider the Bisquare robust weighting function, for which $g(r)$ is defined as follows:

$$g(r) = \begin{cases} \left(1 - (r/h)^2\right)^2 & r \leq h \\ 0 & r > h \end{cases},$$

with h a constant parameter that is fixed in advance (for more details, see [4]). Substituting (5) into (2) gives:

$$\hat{\mathbf{X}}(\mathbf{p}) = \frac{\sum_{\mathbf{q} \in \delta} g \left(\sum_{(\Delta x, \Delta y) \in [-B, \dots, B]^2} \left\| \mathbf{r}_{\mathbf{p}, \mathbf{q}}^{(\Delta x, \Delta y)} \right\|^2 \right) \mathbf{Y}(\mathbf{p} + \mathbf{q})}{\sum_{\mathbf{q} \in \delta} g \left(\sum_{(\Delta x, \Delta y) \in [-B, \dots, B]^2} \left\| \mathbf{r}_{\mathbf{p}, \mathbf{q}}^{(\Delta x, \Delta y)} \right\|^2 \right)}. \quad (6)$$

Comparing (6) to (3) immediately leads to the kernel function:

$$\mathbf{f}_{\mathbf{U}_1^{(1)}}^{(1)}(\mathbf{p}) = \frac{\sum_{\mathbf{q} \in \delta} g \left(\sum_{(\Delta x, \Delta y) \in [-B, \dots, B]^2} \left\| \mathbf{r}_{\mathbf{p}, \mathbf{q}}^{(\Delta x, \Delta y)} \right\|^2 \right) \mathbf{U}_1^{(1)}(\mathbf{p} + \mathbf{q})}{\sum_{\mathbf{q} \in \delta} g \left(\sum_{(\Delta x, \Delta y) \in [-B, \dots, B]^2} \left\| \mathbf{r}_{\mathbf{p}, \mathbf{q}}^{(\Delta x, \Delta y)} \right\|^2 \right)}, \quad (7)$$

with $\mathbf{U}_1^{(1)}(\mathbf{p}) = \mathbf{Y}(\mathbf{p})$. We see that the number of operations performed by the kernel function is linear in $|\delta|(2B + 1)^2$, with $|\delta|$ the cardinality of δ . Although this approach seems feasible, some GPU hardware (especially less recent GPU

hardware) puts limits on the number of operations (more specifically, processor instructions) performed by a kernel function. To work around this restriction, we make use of a weight accumulation buffer (see [19]) and convert every term of the summations in (7) into a separate pass, in which in each pass, one term of the summation $\sum_{\mathbf{q} \in \delta}$ is added to the accumulation buffer. This is done for both the numerator and denominator of (7). For $i = 1, \dots, |\delta|$, with constants \mathbf{q}_i defined for each pass (e.g. using raster scanning), we obtain the kernel function:

$$\mathbf{f}_{U_1^{(i)}, \dots, U_3^{(i)}}^{(i)}(\mathbf{p}) = \begin{pmatrix} U_1^{(i)}(\mathbf{p}) \\ U_2^{(i)} + g \left(\sum_{(\Delta x, \Delta y) \in [-B, \dots, B]^2} \left\| \mathbf{r}_{\mathbf{p}, \mathbf{q}_i}^{(\Delta x, \Delta y)} \right\|^2 \right) U_1^{(i)}(\mathbf{p} + \mathbf{q}_i) \\ U_3^{(i)} + g \left(\sum_{(\Delta x, \Delta y) \in [-B, \dots, B]^2} \left\| \mathbf{r}_{\mathbf{p}, \mathbf{q}_i}^{(\Delta x, \Delta y)} \right\|^2 \right) \end{pmatrix} \quad (8)$$

where $U_2^{(i)}$ is an accumulation buffer for the denoised image, and where $U_3^{(i)}$ is a weight accumulation buffer (initially, $U_2^{(1)}(\mathbf{p}) = U_3^{(1)}(\mathbf{p}) = \mathbf{0}$). Next, one last pass is required, to compute the final output image:

$$\mathbf{f}_{U_1^{(I)}, \dots, U_3^{(I)}}^{(I)}(\mathbf{p}) = \begin{pmatrix} U_2^{(I)}(\mathbf{p}) \\ U_3^{(I)}(\mathbf{p}) \end{pmatrix}^T \mathbf{0}, \quad (9)$$

with $I = |\delta| + 1$. The number of operations per pass is now multiplied by a factor $1/|\delta|$, but is still very high. To further reduce this number of operations, we could apply a similar split-up technique and convert the summation $\sum_{(\Delta x, \Delta y) \in [-B, \dots, B]^2}$ into several passes. We note that, even though this way we would obtain a working algorithm for most available GPUs, the number of passes $I = |\delta|((2B + 1)^2 + 1) + 1$ becomes very high. For example, for a $31 \times 31 \times 4$ -search window and $B = 4$, we obtain $I = 311365$ passes. If for each video frame, a single pass of the algorithm would take 0.1 msec. on a GPU, the complete algorithm would still require approx. 31 sec. for processing one single frame of a video sequence, which is similar to the computation time of our CPU version mentioned in Section 1. Hence, further algorithmic accelerations are required.

2.3 Actual Implementation Using Algorithmic Accelerations

In [4], we pointed out that the term $\sum_{(\Delta x, \Delta y) \in [-B, \dots, B]^2} \left\| \mathbf{r}_{\mathbf{p}, \mathbf{q}}^{(\Delta x, \Delta y)} \right\|^2$ can be interpreted as a convolution operator with a filter kernel with square support. Consequently the Euclidean distance between two patches can efficiently be computed using a moving average filter, and the algorithmic complexity is reduced with roughly a factor $(2B + 1)^2/2$. Unfortunately, converting a moving average filter directly into a GPU program as in (4) is not feasible in a small number of passes. Instead, we exploit the separability of the filter kernel and we implement

the convolution operator as a cascade of a horizontal and vertical filter. Then by setting $\mathbf{U}_1^{(1)}(\mathbf{p}) = \mathbf{Y}(\mathbf{p})$, $\mathbf{U}_2^{(1)}(\mathbf{p}) = \mathbf{U}_3^{(1)}(\mathbf{p}) = \mathbf{U}_4^{(1)}(\mathbf{p}) = \mathbf{0}$, the first pass of our algorithm is as follows:

$$\mathbf{f}_{\mathbf{U}_1^{(4i-3)}, \dots, \mathbf{U}_4^{(4i-3)}}^{(4i-3)}(\mathbf{p}) = \begin{pmatrix} \mathbf{U}_1^{(4i-3)}(\mathbf{p}) \\ \mathbf{U}_2^{(4i-3)}(\mathbf{p}) \\ \mathbf{U}_3^{(4i-3)}(\mathbf{p}) \\ \left\| \mathbf{r}_{\mathbf{p}, \mathbf{q}_i}^{(0,0)} \right\|^2 \end{pmatrix}. \quad (10)$$

Note that the values $\mathbf{U}_1^{(4i-3)}(\mathbf{p})$, $\mathbf{U}_2^{(4i-3)}(\mathbf{p})$, $\mathbf{U}_3^{(4i-3)}(\mathbf{p})$ are simply passed to the next step of the algorithm. We only compute the Euclidean distance between two pixel intensities (in RGB color space). The next passes are given by:

$$\mathbf{f}_{\mathbf{U}_1^{(4i-2)}, \dots, \mathbf{U}_4^{(4i-2)}}^{(4i-2)}(\mathbf{p}) = \begin{pmatrix} \mathbf{U}_1^{(4i-2)}(\mathbf{p}) \\ \mathbf{U}_2^{(4i-2)}(\mathbf{p}) \\ \mathbf{U}_3^{(4i-2)}(\mathbf{p}) \\ \sum_{\Delta x \in [-B, \dots, B]} \mathbf{U}_4^{(4i-2)}(p_x + \Delta x, p_y, p_t) \end{pmatrix},$$

$$\mathbf{f}_{\mathbf{U}_1^{(4i-1)}, \dots, \mathbf{U}_4^{(4i-1)}}^{(4i-1)}(\mathbf{p}) = \begin{pmatrix} \mathbf{U}_1^{(4i-1)}(\mathbf{p}) \\ \mathbf{U}_2^{(4i-1)}(\mathbf{p}) \\ \mathbf{U}_3^{(4i-1)}(\mathbf{p}) \\ g \left(\sum_{\Delta y \in [-B, \dots, B]} \mathbf{U}_4^{(4i-1)}(p_x, p_y + \Delta y, p_t) \right) \end{pmatrix}. \quad (11)$$

The separable filtering reduces the computation complexity by a factor $(2B + 1)/2$. Fortunately, the steps (10) are computationally simple and only require a small number regular memory accesses, which can benefit from the internal memory caches of the GPU. Note that in the last step of (10), we already computed the similarity weights, by evaluating the function $g(\cdot)$.

A second acceleration technique we presented in [19], is to exploit the symmetry property of the weights, i.e. $w(\mathbf{p}, \mathbf{p} + \mathbf{q}_i) = w(\mathbf{p} + \mathbf{q}_i, \mathbf{p})$. To do so, when adding $w(\mathbf{p}, \mathbf{p} + \mathbf{q}_i)\mathbf{Y}(\mathbf{p} + \mathbf{q}_i)$ to the image accumulation buffer at position \mathbf{p} , we proposed to additionally add $w(\mathbf{p}, \mathbf{p} + \mathbf{q}_i)\mathbf{Y}(\mathbf{p})$ to the image accumulation buffer at position $\mathbf{p} + \mathbf{q}_i$. Consequently, the weight $w(\mathbf{p}, \mathbf{p} + \mathbf{q}_i)$ only needs to be computed *once*, effectively halving the size of the search window δ . However, this acceleration technique requires “non-regular” writes to the accumulation buffer, i.e., at position $\mathbf{p} + \mathbf{q}_i$ instead of \mathbf{p} as required by the structure of our GPU program (4). Fortunately, our specific notation here brings a solution here: by noting that \mathbf{q}_i is constant in each pass, we could simply translate the input coordinates and perform a “regular” write to the accumulation buffer. This way, we need to add $w(\mathbf{p} - \mathbf{q}_i, \mathbf{p})\mathbf{Y}(\mathbf{p} - \mathbf{q}_i)$ to the accumulation buffer at position \mathbf{p} . We will call this the *translation* technique. This gives us the next step of our GPU algorithm:

$$\mathbf{f}_{\mathbf{U}_1^{(4i)}, \dots, \mathbf{U}_4^{(4i)}}^{(4i)}(\mathbf{p}) = \left(\begin{array}{c} \mathbf{U}_1^{(4i)}(\mathbf{p}) \\ \mathbf{U}_2^{(4i)} + \mathbf{U}_4^{(4i)}(\mathbf{p})\mathbf{U}_1^{(4i)}(\mathbf{p} + \mathbf{q}_i) + \mathbf{U}_4^{(4i)}(\mathbf{p} - \mathbf{q}_i)\mathbf{U}_1^{(4i)}(\mathbf{p} - \mathbf{q}_i) [1 - \delta(\mathbf{q}_i)] \\ \mathbf{U}_3^{(4i)} + \mathbf{U}_4^{(4i)}(\mathbf{p}) + \mathbf{U}_4^{(4i)}(\mathbf{p} - \mathbf{q}_i) [1 - \delta(\mathbf{q}_i)] \\ \mathbf{U}_4^{(4i)}(\mathbf{p}) \end{array} \right) \quad (12)$$

with $\delta(\cdot)$ the Dirac delta function. The Dirac delta function is needed here, to prevent the weights $w(\mathbf{p}, \mathbf{p})$ to be counted twice. In the last pass, again the image accumulation buffer intensities are divided by the accumulated weights, which gives:

$$\mathbf{f}_{\mathbf{U}_1^{(I)}, \dots, \mathbf{U}_4^{(I)}}^{(I)}(\mathbf{p}) = \left(\frac{\mathbf{U}_2^{(I)}(\mathbf{p})}{\mathbf{U}_3^{(I)}(\mathbf{p})} \mathbf{0} \mathbf{0} \mathbf{0} \right)^T, \quad (13)$$

with $I = 4(|\delta| + 1)/2 + 1 = 2|\delta| + 3$. The output of the NLMeans algorithm is then $\hat{\mathbf{X}}(\mathbf{p}) = \mathbf{U}_2^{(I)}(\mathbf{p})/\mathbf{U}_3^{(I)}(\mathbf{p})$. Consequently, the complete NLMeans algorithm comprises the passes $i = 1, \dots, I$ defined by steps (10)-(13).

2.4 Extension to Noise Correlated across Color Channels

In this Section, we briefly explain how our GPU-NLMeans algorithm can be extended to deal with Gaussian noise that is correlated across color channels. Our main goal here is to show that our video algorithm is not restricted to white Gaussian noise. Because of space limitations, visual and quantitative results for color images and color video will be reported in later publications. As we pointed out in [4, p. 6], the algorithm can be extended to spatially correlated noise by using a Mahalanobis distance based on the noise covariance matrix instead of the Euclidean distance similarity metric. When dealing with noise which is correlated across color channels, we need to replace (5) by:

$$w(\mathbf{p}, \mathbf{p} + \mathbf{q}) = g \left(\sum_{(\Delta x, \Delta y) \in [-B, \dots, B]^2} \left(\mathbf{r}_{\mathbf{p}, \mathbf{q}}^{(\Delta x, \Delta y)} \right)^T \mathbf{C}^{-1} \left(\mathbf{r}_{\mathbf{p}, \mathbf{q}}^{(\Delta x, \Delta y)} \right) \right)$$

with \mathbf{C} the noise covariance function. In practice, the matrix \mathbf{C} can be estimated from flat regions in the video sequence, or based on an EM-algorithm as in [27]. Now, by introducing the decorrelating color transform $\mathbf{G} = \mathbf{C}^{-1/2}$, and by defining:

$$\mathbf{r}'_{\mathbf{p}, \mathbf{q}}^{(\Delta x, \Delta y)} = \mathbf{G}\mathbf{Y}(p_x + q_x + \Delta x, p_y + q_y + \Delta y, p_y + q_y) - \mathbf{G}\mathbf{Y}(p_x + \Delta x, p_y + \Delta y, p_t),$$

the weighting function can again be expressed in terms of the Euclidean distance $\left\| \mathbf{r}'_{\mathbf{p}, \mathbf{q}}^{(\Delta x, \Delta y)} \right\|^2$. Hence, removing correlated noise from video sequences solely requires a color transform \mathbf{G} applied as pre-processing to the video sequence. Furthermore, this technique can be combined with our previous approach from [4, p. 6] in order to remove Gaussian noise which is both spatially correlated and correlated across color channels.

2.5 Discussion

To optimize the computational performance of a GPU program, minimizing the number of passes I and performing more operations in each kernel function is more beneficial than optimizing the individual kernel functions themselves, especially when the kernel functions are relatively simple (as in our algorithm in Section 2.3). This is due to GPU memory caching behavior and also because every pass typically requires interaction with the CPU (for example, the computation time of an individual pass can be affected by the process CPU scheduling granularity). To assess the computational performance improvement, a possible solution would be to use theoretical models to predict the performance. Unfortunately, these theoretical models are very dependent on the underlying GPU architecture: the computational performance can not simply be expressed as a function of the total number of floating point operations, because of the parallel processing. To obtain a rough idea of the computational performance we use the actual number of passes required by our algorithm. For example, when comparing our algorithmic accelerations from Section 2.3 to the naive NLMeans-algorithm from Section 2.2, we see that the number of passes is reduced with a factor:

$$\frac{|\delta| \left((2B + 1)^2 + 1 \right) + 1}{4 \left(|\delta| + 1 \right) / 2 + 1} \approx \frac{(2B + 1)^2}{2}.$$

For patches of size 9×9 , the accelerated NLMeans GPU algorithm requires approximately 40 times less processing passes.

Another point of interest is the streaming behavior of the algorithm: for real-time applications, it is required the algorithm processes video frames as soon as they become available. In our algorithm, this can be completely controlled by adjusting the size of the search window. Suppose we choose:

$$\delta = [-A, \dots, A] \times [-A, \dots, A] \times [-D_{\text{past}}, \dots, D_{\text{future}}]$$

with $A, D_{\text{past}}, D_{\text{future}} \geq 0$ positive constants. A determines the size of the spatial search window; D_{past} and D_{future} are respectively the number of past and future frames that the filter uses for denoising the current frame. For causal implementation of the filter, a delay of D_{future} frames is required. Of course, D_{future} can even be zero, if desired. However, the main disadvantage of a zero delay is that the translation technique from Section 2.3 cannot be used in the temporal direction, because the translation technique in fact requires the updating of future frames in the accumulation buffer. Nevertheless, using a small positive D_{future} , a trade-off can be made between the filter delay and the algorithmic acceleration achieved by exploiting the weight symmetry. The number of video frames in GPU memory is at most $4(D_{\text{past}} + D_{\text{future}} + 1)$.

3 Experimental Results

To demonstrate the processing time improvement of our GPU algorithm with the proposed accelerations, we apply our technique to a color video sequence of resolution 720×480 (a resolution which is common for DVD-video). The video sequence is corrupted with artificially added stationary white Gaussian noise with

standard deviation 25/255 (input PSNR 20.17dB). We compare the processing time of our proposed GPU implementation to the modestly optimized (single-threaded) C++ CPU implementation from our previous work [4] (including all acceleration techniques proposed in [4]), for different values of the parameters A and D_{past} . For these results, we use $D_{\text{future}} = 0$ (resulting in a zero-delay denoising filter, as explained in Section 2.5), $B = 4$ (corresponding to 9×9 patches) and we manually select h to optimize the PSNR ratio. In particular, we use $h = 0.13$ for $A \leq 3$ and $h = 0.16$ for $A > 3$ (note that the pixel intensities are within the range $0 - 1$).

Both the CPU and GPU version were run on the same computer, which is equipped with a 2.4GHz Intel Core(2) processor with 2048 MB RAM and a NVidia GeForce 9600GT GPU. This card has 64 parallel stream processing units and is considered to be a mid-range GPU. The GPU algorithm is implemented as a HLSL pixel shader in DirectX 9.1 (Windows XP) and makes use of 16-bit floating point values. The main program containing the GPU host code, is written in C# 3.0.

Processing time and output PSNR results (obtained after denoising) are reported in Table 1. We only report PSNR results for the GPU denoising technique, since both CPU and GPU algorithms essentially compute the same formula (i.e. equation (2)). It can be seen that the PSNR values increase when using a larger search window or a larger number of past frames. This is simply because more similar candidate blocks become available for searching, and consequently bet-

Table 1. Experimental results for denoising a color video sequence, consisting of 99 frames of dimensions 720×480 and corrupted with additive stationary white Gaussian noise with standard deviation 25/255 (PSNR=20.17dB)

Parameters			GPU			CPU	GPU vs. CPU
A	Search window	D_{past}	FPS	msec/frame	PSNR [dB]	msec/frame	acceleration
2	5x5	0	100.00	10.00	33.09	4021	402.10×
2	5x5	1	69.57	14.37	34.42	N/A	
2	5x5	2	50.79	19.69	34.88	N/A	
2	5x5	3	40.34	24.79	35.04	N/A	
3	7x7	0	52.46	19.06	35.40	7505	393.70×
3	7x7	1	30.38	32.92	36.19	N/A	
3	7x7	2	21.43	46.67	36.26	N/A	
3	7x7	3	16.58	60.31	36.33	N/A	
5	11x11	0	18.46	54.17	36.34	18230	336.55×
5	11x11	1	10.22	97.81	37.11	N/A	
5	11x11	2	7.07	141.35	37.26	N/A	
5	11x11	3	5.42	184.48	37.23	N/A	
10	21x21	0	4.32	231.56	36.79	50857	219.63×
10	21x21	1	2.36	424.27	37.20	N/A	
10	21x21	2	1.62	615.73	37.24	N/A	
10	21x21	3	1.24	805.83	37.17	N/A	

ter estimates can be found for the denoised pixel intensities. Remarkable is also the huge acceleration of the GPU compared to the CPU of a factor 200 to 400. The main reason lies in the massive amount of parallelism in the NLMeans algorithm, which can be fully exploited by the GPU but hardly by the CPU. Especially this huge acceleration leads to a real-time denoising filter. We can determine the optimal parameters for the algorithm by selecting a minimum frame rate and by maximizing the output PSNR of the filter for this minimum frame rate. For our results in Table I, an optimal combination is a 7×7 -search window and $D_{\text{past}} = 1$, in order to attain a frame rate of 25 frames per second (fps).

4 Conclusion

In this paper, we have shown how the traditional NLMeans algorithm can be efficiently mapped onto a parallel processing architecture such as the GPU. We saw that a naive straightforward implementation inevitably leads to an inefficient algorithm with a huge number of parallel processing passes. We then analyzed our NLMeans algorithmic acceleration techniques from previous work, and we noted that these techniques can not be applied “as is”. Therefore, we adapted the core ideas of these acceleration techniques (i.e. the moving averaging filter for the fast computation of Euclidean distances and the exploitation of the weight symmetry) to GPGPU programming methodology and we arrived at a GPU-NLMeans algorithm that is two to three orders of magnitudes faster (depending on the parameter choices) than the equivalent CPU algorithm. This technique can process video sequences in real-time on a mid-range GPU.

References

1. Rudin, L., Osher, S.: Total variation based image restoration with free local constraints. In: IEEE Int. Conf. Image Proc (ICIP), vol. 1, pp. 31–35 (November 1994)
2. Portilla, J., Strela, V., Wainwright, M., Simoncelli, E.P.: Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Trans. Image Processing* 12(11), 1338–1351 (2003)
3. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3d transform-domain collaborative filtering. *IEEE Trans. Image Processing* 16(8), 2080–2095 (2007)
4. Goossens, B., Luong, H., Pižurica, A., Philips, W.: An improved Non-Local Means Algorithm for Image Denoising. In: Int. Workshop on Local and Non-Local Approx. in Image Processing (2008) (invited paper)
5. Goossens, B., Pižurica, A., Philips, W.: Removal of correlated noise by modeling the signal of interest in the wavelet domain. *IEEE Trans. Image Processing* 18(6), 1153–1165 (2009)
6. Goossens, B., Pižurica, A., Philips, W.: Image Denoising Using Mixtures of Projected Gaussian Scale Mixtures. *IEEE Trans. Image Processing* 18(8), 1689–1702 (2009)
7. Brailean, J.C., Kleihorst, R.P., Efstraditis, S., Katsageleos, K.A., Lagendijk, R.L.: Noise reduction filters for dynamic image sequences: a review. *Proc. IEEE* 83(9), 1272–1292 (1995)

8. Selesnick, I.W., Li, K.Y.: Video denoising using 2D and 3D dual-tree complex wavelet transforms. In: Proc. SPIE Wavelet Applications in Signal and Image Processing, pp. 607–618 (August 2003)
9. Pižurica, A., Zlokolica, V., Philips, W.: Combined wavelet domain and temporal video denoising. In: Proc. IEEE Int. Conf. on Advanced Video and Signal Based Surveillance (AVSS), pp. 334–341 (2003)
10. Zlokolica, V., Pižurica, A., Philips, W.: Recursive temporal denoising and motion estimation of video. In: IEEE Int. Conf. Image Proc (ICIP), pp. 1465–1468 (2004)
11. Goossens, B., Pižurica, A., Philips, W.: Video denoising using motion-compensated lifting wavelet transform. In: Proceedings of Wavelets and Applications Semester and Conference (WavE 2006), Lausanne, Switzerland (July 2006)
12. Dabov, K., Foi, A., Egiazarian, K.: Video denoising by sparse 3D transform-domain collaborative filtering. In: European Signal Processing Conference (EUSIPCO 2007), Poznan, Poland (2007)
13. Buades, A., Coll, B., Morel, J.-M.: Nonlocal Image and Movie Denoising. *Int J. Comput. Vis.* 76, 123–139 (2008)
14. Yu, S., Ahmad, M.O., Swamy, M.N.S.: Video Denoising using Motion Compensated 3D Wavelet Transform with Integrated Recursive Temporal Filtering. *IEEE Trans. Cir. and Sys. for Video Technol.* (2010) (in press)
15. Mélangé, T., Nachttegaal, M., Kerre, E.E., Zlokolica, V., Schulte, S., De Witte, V., Pizurica, A., Philips, W.: Video denoising by fuzzy motion and detail adaptive averaging. *Journal of Elec. Imaging* 17(4), 43005–1–43005–19 (2008)
16. Buades, A., Coll, B., Morel, J.M.: A non local algorithm for image denoising. In: Proc. Int. Conf. Comp. Vision and Pat. Recog (CVPR), vol. 2, pp. 60–65 (2005)
17. Azzabou, N., Paragias, N., Guichard, F.: Image Denoising Based on Adapted Dictionary Computation. In: Proc. of IEEE International Conference on Image Processing (ICIP), San Antonio, Texas, USA, pp. 109–112 (September 2007)
18. Kervrann, C., Boulanger, J., Coupé, P.: Bayesian Non-Local Means Filter, Image Redundancy and Adaptive Dictionaries for Noise Removal. In: Sgallari, F., Murli, A., Paragios, N. (eds.) *SSVM 2007*. LNCS, vol. 4485, pp. 520–532. Springer, Heidelberg (2007)
19. Dauwe, A., Goossens, B., Luong, H.Q., Philips, W.: A Fast Non-Local Image Denoising Algorithm. In: Proc. SPIE Electronic Imaging, San José, USA, vol. 6812 (January 2008)
20. Kervrann, C., Boulanger, J.: Optimal spatial adaptation for patch-based image denoising. *IEEE Trans. Image Processing* 15(10), 2866–2878 (2006)
21. Wang, J., Guo, Y., Ying, Y., Liu, Y., Peng, Q.: Fast non-local algorithm for image denoising. In: IEEE Int. Conf. Image Proc (ICIP), pp. 1429–1432 (2006)
22. Bilcu, R.C., Vehvilainen, M.: Fast nonlocal means for image denoising. In: Martin, R.A., DiCarlo, J.M., Sampat, N. (eds.) *Proc. SPIE Digital Photography III*, vol. 6502, SPIE, CA (2007)
23. Aelterman, J., Goossens, B., Pižurica, A., Philips, W.: Suppression of Correlated Noise, IN-TECH. In: *Recent Advances in Signal Processing* (2010)
24. General-Purpose Computation on Graphics Hardware, <http://www.gpgpu.org>
25. Kharlamov, A., Podlozhnyuk, V.: Image denoising, CUDA 1.1 SDK (June 2007)
26. De Fontes, F.P.X., Barroso, G.A., Hellier, P.: Real time ultrasound image denoising. *Journal of Real-Time Image Processing* (April 2010)
27. Goossens, B., Pižurica, A., Philips, W.: EM-Based Estimation of Spatially Variant Correlated Image Noise. In: IEEE Int. Conf. Image Proc. (ICIP), San Diego, CA, USA, pp. 1744–1747 (2008)

An Efficient Mode Decision Algorithm for Combined Scalable Video Coding

Tae-Jung Kim, Bo-Seok Seo, and Jae-Won Suh

Chungbuk National University, College of Electrical and Computer Engineering,
12 Gaeshin-dong, Heungduk-gu, Chongju, Korea
taejung@cbnu.ac.kr, boseok@cbnu.ac.kr, sjwon@cbnu.ac.kr

Abstract. Scalable video coding (SVC) is an extension of H.264/AVC that is used to provide a video standard for scalability. Scalability refers to the capability of recovering physically meaningful image or video information by decoding only partial compressed bitstreams. Scalable coding is typically accomplished by providing multiple layers of a video, in terms of quality resolution, spatial resolution, temporal resolution, or combinations of these options. To increase the coding efficiency, SVC adapts the inter layer prediction which uses the information of base layer to encode the enhancement layers. Due to the inter layer prediction, the computational complexity of SVC is much more complicated than that of H.264/AVC, such as mode decision based on rate-distortion optimization (RDO) and hierarchical bi-directional motion estimation. In this paper, we propose a fast mode decision algorithm for combined scalability to reduce the complexity. Experimental results show that the proposed algorithm achieves up to a 48% decrease in the encoding time with a negligible loss of visual quality and increment of bit rates.

Keywords: Scalable video coding, Fast mode decision, Combined scalability, H.264/AVC SE.

1 Introduction

The communication channels comprising a modern network span a broad bandwidth range. Therefore, the compressed bitstreams created for particular resource may not be satisfactory, efficient, or useful for servicing users with different resource capacities. To support these flexible requirements, SVC has been adopted as an amendment to H.264/AVC [1] and finalized as an extension to H.264/AVC video standard [2]. SVC simultaneously generates single base layer and several enhancement layers during the encoding procedure. The basic coding information is encoded as a base layer with reduced resolution, frame rate, and quality, which can be used for mobile devices. The enhancement layers supported by base layer provide a high quality service.

To increase the coding efficiency, H.264/AVC adapts several advanced coding techniques, such as mode decision for macroblock (MB) coding, 4×4 integer discrete cosine transform (DCT), content adaptive binary arithmetic coding

(CABAC), etc [1]. Especially, mode decision including spatial prediction for intra mode and variable block size motion estimation/compensation (ME/MC) with multiple reference frames for inter mode is much more complicated for SVC because the inter layer prediction is used between layers. Therefore, it is necessary to design a method to reduce this complexity with a minimal loss of image quality.

Many kinds of fast mode decision schemes have been proposed for H.264/AVC: Fast variable block size motion estimation (ME) [3], fast inter coding mode selection [4][5][6], fast intra prediction [7], etc. Recently, some fast mode decision algorithms for SVC have been reported. However, almost every fast algorithm for SVC has been specialized in a single scalability. A fast mode decision algorithm for spatial scalability has been suggested by Li et al. [8] in which the mode distribution relationship between the base layer and enhancement layers is used. Lim et al. [9] proposed a fast encoding mode decision method using an early skip mode detection technique based on the relationship between the temporal levels in a group of pictures (GOP). Some literatures try to combine scalability [10][11]. A fast mode decision algorithm by Li et al. [10] can support partially combined scalability: spatial scalability, a coarse grain signal-to-noise ratio (CGS), and temporal scalability. They use the correlation of mode distribution between the base layer and enhancement layers. A layer adaptive mode decision algorithm and a motion search scheme by Lin et al. [11] have been proposed for CGS and temporal scalability in which modes with limited contributions to the coding efficiency are skipped based on a statistical analysis in order to reduce the computational complexity of the mode search.

In this paper, we propose a fast mode determination scheme for inter frame coding supporting temporal, spatial, and quality scalability based on correlative information between the base layer and enhancement layers. In the proposed algorithm, we define a cost function for the motion area based on ordered mode information and two mode search classes: large block type (16×16 , 16×8 , 8×16) and sub block type (8×8 , 8×4 , 4×8 , 4×4). Based on the designed cost function for the motion area, either large block type or sub block type is assigned to the mode search process. Next, we determine the direction of mode search (forward, backward, bidirectional) using the direction of first mode type for the determined block type. Using this direction, mode search for the remaining modes in the block type is performed to find the best mode type for the current macroblock (MB).

2 Inter Frame Coding for Combined SVC

SVC supports three special types of scalability that allow complete images to be decoded from only part of the bitstream. The three types are spatial, temporal, and quality. In addition, each scalability can be combined to support general condition. Because SVC is an extension of H.264/AVC, most advanced coding techniques for H.264/AVC are used for inter frame coding. In this section, we briefly introduce the type of scalability and the mode decision process.

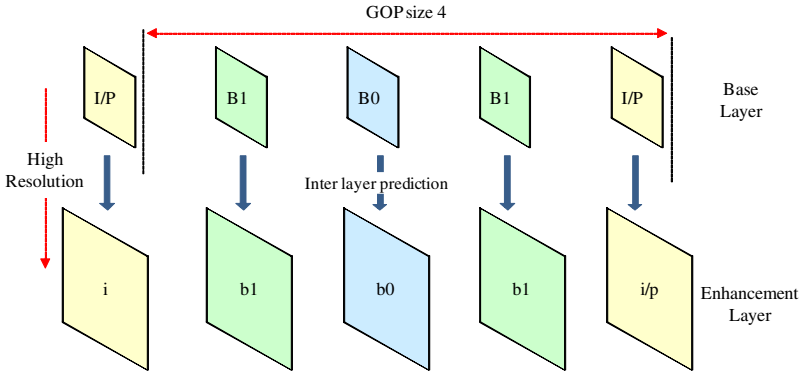


Fig. 1. The structure of the spatial scalability with GOP size 4

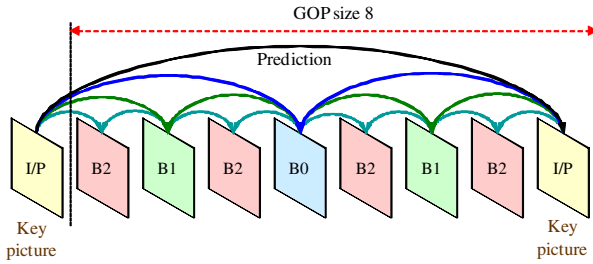


Fig. 2. Bi-directional prediction for hierarchical B pictures

2.1 Type of Scalability

Spatial Scalability. As shown in Fig. 1, it generally starts with a base layer at a lower resolution and adds an enhancement layer at higher resolution. The input video source for base layer is preprocessed to create the lower resolution image and is independently coded with H.264/AVC. In the enhancement layer the difference between an interpolated version of the base layer and the source image are coded. However, both base layer and enhancement layer have the same frame rate and quality.

Quality Scalability. Quality scalability maintains the same luminance resolution and frame rate in the lower layer and a single enhancement layer. However, different quantization scales support the different qualities.

Temporal Scalability. Temporal scalability, provided by the hierarchical B picture structure, uses a technique to encode different temporal resolutions with the same spatial resolution as shown in Fig. 2. The hierarchical B picture structure can be made using bi-directional motion prediction in the GOP. The bi-directional motion search uses both forward reference pictures (list 0) and backward reference pictures (list 1). As depicted in Fig. 2, the picture at the lowest

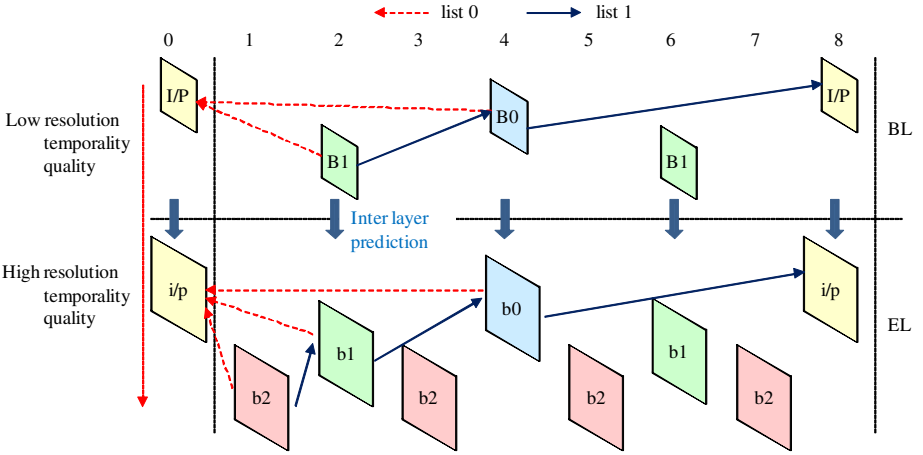


Fig. 3. The structure of combined scalability

level is called the key picture and is encoded as an intra (I-picture) or a predictive frame (P-picture).

Combined Scalability. These types of scalability can be combined. The structure of combined scalability has features of both spatio-temporal and quality scalability. For example, Fig. 3 shows the encoding structure for the combined scalability with two layers. The base layer is encoded at a lower resolution, coarse quality, and a slow temporal rate. The enhancement layer is encoded at a higher resolution, fine quality, and a faster temporal rate.

The enhancement layer is also divided into an inter layer prediction picture (i/p, b0, and b1) and a non-inter layer prediction picture (b2). To increase the coding efficiency, the inter layer prediction picture can use the encoded information of the base layer, such as intra texture, motion vector, and residual coefficients. The non-inter layer picture is only encoded by using bi-directionally adjacent frames in the same layer.

2.2 Mode Decision in SVC

H.264/AVC uses the RD optimization (RDO) technique to determine the best MB coding mode in terms of minimizing bit rates and maximizing image quality.

Inter Mode Prediction. Unlike the inter MB mode of previous video coding standards, MB for H.264/AVC can be motion estimated and compensated from already transmitted multi reference frames with varying block sizes from 16×16 down to 4×4 as shown in Fig. 4. One of these various types is determined as the best inter mode. To determine the best inter MB mode in terms of minimizing bit rate and maximizing image quality, the RDO technique is used. The best inter MB mode is determined as one having the smallest RD cost in Eq. (II).

$$J_{inter} = SSD\{s, r(MV)\} + \lambda_{motion} \cdot R(MVD, REF), \quad (1)$$

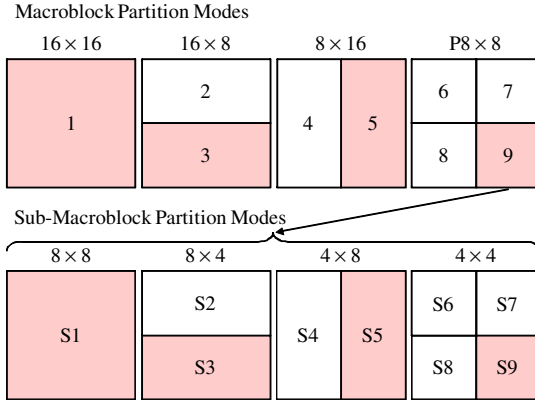


Fig. 4. Inter Prediction Modes

where s and $r(MV)$ mean the current block and the predicted block with the estimated MV , respectively. $SSD\{s, r(MV)\}$ is the sum of squared differences between the current block s and its corresponding block r . λ_{motion} is the Lagrange multiplier and $R(MVD, REF)$ means the bit rates for encoding the motion vector difference MVD and the number of reference frame REF .

Intra Mode Prediction. In addition to the inter MB coding types, various intra prediction modes are specified in H.264/AVC as shown in Fig. 5. Unlike the previous coding standards, intra prediction in H.264/AVC is performed in the pixel domain by referring to neighboring samples. First, prediction blocks are formed by directional interpolation using the intra prediction modes. Next, we calculate the RD cost of the difference between the current MB and its corresponding prediction block. The best intra mode is determined as one having the smallest RD cost. The RD cost function is

$$J_{intra} = SSD\{s, r\} + \lambda_{mode} \cdot R(s, r, M), \tag{2}$$

where λ_{mode} is the Lagrange multiplier and $R(s, r, M)$ means the bit rates for encoding the residual data according to the predicted mode M .

The Best Coding Mode Selection. To obtain the best MB coding mode for the P-frame, H.264/AVC encoder exhaustively tests all possible encoding modes including inter modes and intra modes. As a result, the mode having the smallest RD cost is determined the best coding MB mode among all possible modes in the P-frame.

The Problem of Mode Decision for Combined SVC. A simple structure for combined SVC with inter layer prediction is shown in Fig. 3. When the inter layer prediction is off, the multi-layer signals are equal to multiple independent sequences transmission. In order to improve the encoding efficiency, the inter

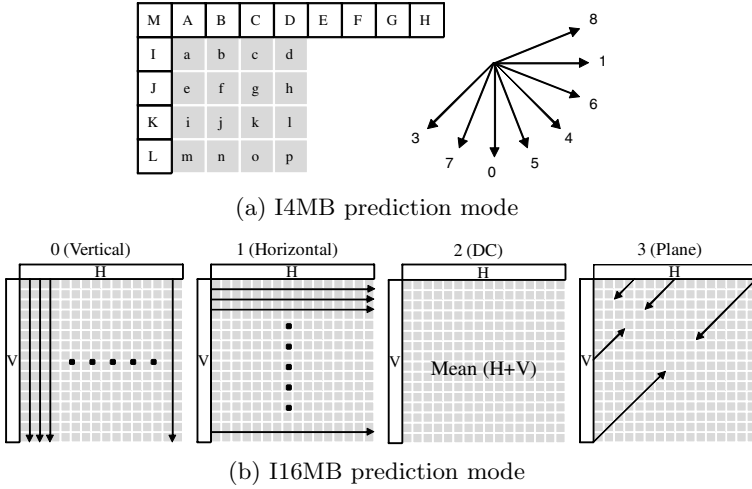


Fig. 5. Intra Prediction Modes

layer prediction is normally set to adaptive mode. With this condition, the inter layer prediction signal is either formed by motion compensated prediction inside the same layer or by up-sampled reconstructed lower layer signal. Adaptive inter layer prediction chooses the best mode using the RDO function with tremendous coding complexity. Therefore, we need a fast mode decision algorithm to reduce the complexity with a negligible loss of visual quality and increment of bit rates.

3 Proposed Mode Decision Algorithm for Combined SVC

We concentrate on motion area to develop the fast mode decision algorithm. We first define a cost for the motion area and then the directional mode search is applied using this cost.

3.1 Cost for Motion Area

Fig. 6 shows the structure of combined scalability which is composed of two layers that have different frame rates, spatial resolutions, and quantization scales. The base layer is encoded at a lower frame rate, lower resolution, and quality while the enhancement layer uses a higher frame rate, larger resolution, and better quality.

The cost for the motion area (MA_{cost}) is defined first. MA_{cost} , the predicted complexity for the current MB, is described as a degree of correlation between the base and enhancement layers. The cost is expressed as

$$MA_{cost} = Temporal_{cor} + Quality_{cor} \cdot Spatial_{cor}, \quad (3)$$

where cor is the degree of correlation. $Temporal_{cor}$ indicates a relationship between the lower and higher temporal levels. $Quality_{cor}$ and $Spatial_{cor}$ are defined by a relationship between the base layer and the enhancement layer.

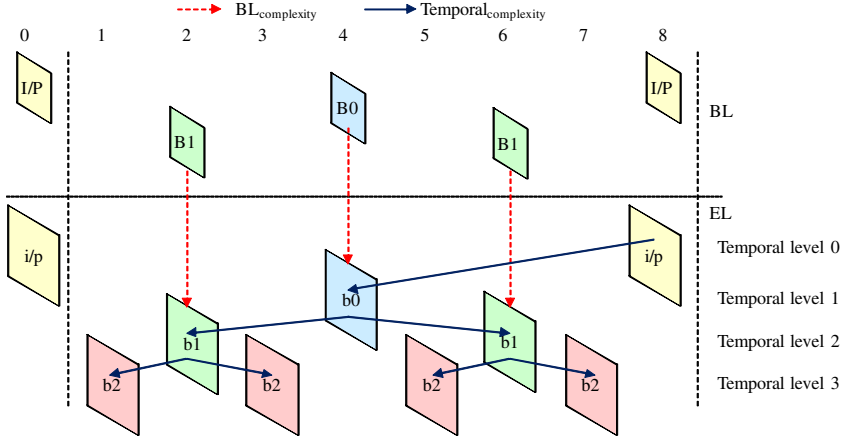


Fig. 6. The use of information for combined scalability structure

$Temporal_{cor}$ for the current MB can be calculated from the mode complexity and the motion vector of the previous temporal level at the corresponding MB position. The estimated $Temporal_{cor}$ is calculated by Eq. (4).

$$\begin{aligned}
 Temporal_{cor} &= Temporal_{complexity}^l \\
 &= Mode_{complexity}^{l-1} \times \frac{MV_{val}^{l-1}}{search\ size}, \tag{4}
 \end{aligned}$$

where l is the temporal level as shown in Fig. 6. $Mode_{complexity}$ indicates the proposed mode number in Table 1. The value of $search\ size$ represents the maximum length of search range. MV_{val} is defined by MV values in the corresponding MB of the previous temporal level, which is expressed as

$$MV_{val} = \lceil avg(|MV_x| + |MV_y|)_n \rceil, \tag{5}$$

where n is the number of MV in the corresponding MB. The number of MV can be various because partitioned ME is permitted in H.264/AVC. It is a integer value by round up.

$Quality_{cor} \cdot Spatial_{cor}$ is defined as $BL_{complexity}$, which represents the complexity of texture in the base layer. For spatial scalability, the image in the base layer and the image in the enhancement layer are very similar. However, the visual quality between the base layer and the enhancement layer is extremely different, which is affected by the different quantization scales. Therefore, $BL_{complexity}$ is calculated by

$$\begin{aligned}
 Quality_{cor} \cdot Spatial_{cor} &= BL_{complexity} \\
 &= scale\ factor \times BL_Mode_{complexity} \\
 scale\ factor &= \frac{1}{\log_2(diff_QP)}, \tag{6} \\
 diff_QP &= |QP_{BL} - QP_{EL}|, \quad 2 \leq diff_QP < 51,
 \end{aligned}$$

Table 1. Ordered mode information

Type of mode	Mode number of JSVM	Proposed mode number
<i>SKIP</i>	0	0
16×16	1	1
16×8	2	2
8×16	3	2
8×8	4	4
<i>I4MB</i>	6	6
<i>I16MB</i>	12	6

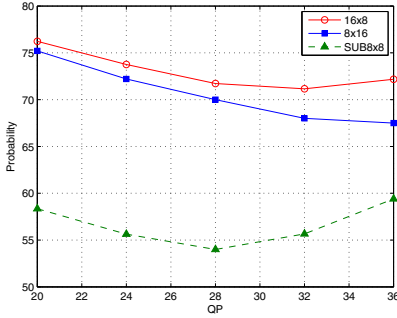
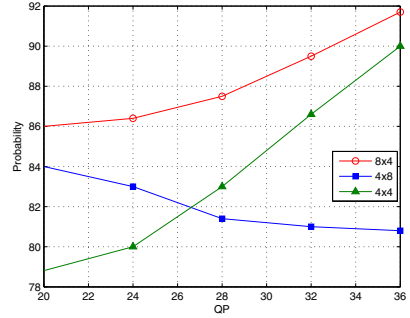
(a) A 16×16 mode and other modes(b) A 8×8 mode and detailed block modes

Fig. 7. Distribution of the conditional probabilities for a search direction between the representative mode and other modes(The average of Foreman, Bus and Mobile sequences)

where $BL_{Mode_{complexity}}$ represents the proposed mode number of the base layer in Table 1, QP_{BL} and QP_{EL} represent the quantization scales in the base layer and enhancement layer, respectively.

3.2 Directional Motion Search

The hierarchical-B picture is constructed using the bi-directional motion search with both forward reference and backward reference pictures as shown in Fig. 6. This technique achieves a high encoding efficiency but it increases the computational complexity of the encoder. Therefore, we propose a directional motion search to reduce the complexity.

We performed an analysis of the characteristics of the bi-directional motion search. Fig. 7 shows the mode correlation between the representative mode (16×16 or 8×8) and other modes. Fig. 7(a) shows the mode correlation between the 16×16 and other mode within MB partition modes (16×16 , 16×8 , 8×16). It means that the search direction for MB partition modes is highly correlated between the current MB and the corresponding MB for previous temporal level. However, the 16×16 search direction has a lower correlation value with the 8×8

search direction. In sub_MB partition modes, we can get the similar results as shown in Fig. 7(b). The search direction for the sub_MB partition modes (8×8 , 8×4 , 4×8 , 4×4) is also highly correlated.

Based on this observation, we can determine the search direction for the current MB that minimize the RD cost between two modes, 16×16 for MB partition modes or 8×8 for sub_MB modes. Using this direction, a further mode search is performed to find the best mode type for the current MB.

Next, we use a feedback loop to prevent a large quality loss. If the best RD cost in the selected class is larger than the defined adaptive threshold ($Mode_{ATh}$), then we go to the other class and perform an additional mode search. The adaptive threshold value $Mode_{ATh}$ is computed as follows using the RD costs of neighboring MBs.

$$Mode_{ATh} = avg(RD_A, RD_B, RD_C \text{ or } B) \quad (7)$$

The position A , B , and C is the same as that of the prediction MV in H.264/AVC.

4 Experimental Results

To verify the performance of the proposed fast mode determination for combined scalability in SVC, simulations were performed on various test sequences using JSVM 9.17 reference software. Table 2 shows the simulation conditions. The spatial resolution and the frame rate in the enhancement layer were twice those of the base layer. The spatial resolution in the enhancement layer had a common intermediate format (CIF) size (FOREMAN, MOBILE, CITY, BUS, SOCCER, and FOOTBALL) and a standard-definition size (ICE and HARBORU)

The measures for evaluating the performance of the proposed algorithm were $BDPSNR$ (dB), $BDBR$ (%) [12], and $\Delta Time$ (%). $\Delta Time$ represents a comparison factor indicating the average for the amount of saved encoding time at each QP.

Table 2. Simulation conditions

layer parameter		Conditions	
QP	Base	40	
	Enhancement	24, 28, 32, 36	
Resolution	Base	QCIF(7)	CIF(2)
	Enhancement	CIF(7)	SD(2)
Frame Rate	Base	15Hz(7)	30Hz(2)
	Enhancement	30Hz(7)	60Hz(2)
Coding Option		MV search: 32 MV resolution: 1 \ 4 pel Reference frame: 1, GOP: 8 Total encoding frame: 97 CAVLC, Loop Filter off	

Table 3. Simulation results for combined scalability

Sequence	BDPSNR (dB)		BDBR (%)		Δ Time(sec)	
	Li's	Ours	Li's	Ours	Li's	Ours
FOREMAN	-0.67	-0.07	14.79	1.60	39.25	48.38
MOBILE	-0.31	-0.05	6.90	1.33	38.94	43.82
CITY	-0.71	-0.04	13.51	0.89	38.32	47.68
BUS	-0.52	-0.10	9.53	1.80	39.13	46.49
SOCCER	-0.61	-0.08	11.07	1.50	38.85	47.94
FOOTBALL	-0.56	-0.23	9.48	3.90	36.91	44.11
ICE	-0.70	-0.05	17.20	1.50	37.89	46.98
HARBOUR	-0.27	-0.03	7.60	1.00	40.52	43.10
Average	-0.54	-0.08	11.26	1.69	38.72	46.06

$$\Delta Time = \frac{Time[reference] - Time[Proposed]}{Time[reference]} \times 100 \quad (8)$$

We used Li's method [10], a well known fast mode decision technique in the SVC encoding system, for an objective comparison of the encoding performance of our algorithm to provide spatial and temporal scalability. Results for Li's method are shown separately [10]. For a comparison with our algorithm, we implemented Li's method to support combined scalability because our proposed algorithm is designed to support combined scalability.

Table 3 shows the simulation results for combined scalability with various QP values. The average loss in *BDPSNR* was measured as -0.23~ 0.03dB and *BDBR* increased 0.89~ 3.9%, compared with the full mode search. The proposed algorithm increases the speed of the SVC encoding system up to 48.38% at FOREMAN, compared to the full mode search. Compared to Li's method, the proposed algorithm achieved a speed-up gain of up to 9% with a smaller bit increment. Li's method resulted in a large quality loss (0.54 (dB)) and a large bit increment (11.26%) for combined scalability. The proposed algorithm resulted in a speed-up gain of approximately 8% more than Li's method while suffering less quality loss and a smaller bit rate increment.

5 Conclusions

A fast mode decision algorithm is proposed for inter-frame encoding with combined scalability. This algorithm is based on correlative information between the base layer and enhancement layers and correlation of temporal levels. In the proposed algorithm, we define a cost for the motion area using ordered mode information. Our scheme also uses two classes for the mode search and a feedback structure to guarantee image quality. For combined scalability, experimental results show that the proposed algorithm significantly reduces the computational complexity of the SVC encoder up to 48% with only a small PSNR loss and bit rate increment.

Acknowledgement

This work was supported by Mid-career Researcher Program through NRF grant funded by the MEST (No. R01-2008-000-20485-0).

References

1. Wiegand, T., Sullivan, G.J., Bjontegard, G., Luthra, A.: Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuit Syst. Video Technol.* 13, 560–576 (2003)
2. Schwarz, H., Marpe, D., Wiegand, T.: Overview of the scalable video coding extension of the H.264/AVC standard. *IEEE Trans. Circuit Syst. Video Technol.* 17, 1103–1120 (2007)
3. Kuo, T.Y., Chan, C.H.: Fast Variable Block Size Motion Estimation for H.264 Using Lilelhood and Correlation of Motion Field. *IEEE Trans. Circuit and Systems for Video Technology* 16, 1185–1195 (2006)
4. Choi, I., Lee, J., Jeon, B.: Fast coding mode selection with rate-distortion optimization for MPEG-4 part-10 AVC/H.264. *IEEE Trans. Circuit and Systems for Video Technology* 16(12), 1557–1561 (2006)
5. Kim, B.J.: Novel inter mode decision algorithm based on macroblock tracking for the P-slice in H.264/AVC video coding. *IEEE Trans. Circuit Syst. Video Technol.* 18, 273–279 (2008)
6. Crecos, C., Yang, M.Y.: Fast inter mode prediction for P slice in the H.264 video coding standard. *IEEE Trans. Broadcasting* 51, 256–263 (2005)
7. Pan, F., Lin, X., Rahardja, S., Lim, K.P., Li, Z.G., Wu, D., Wu, S.: Fast Mode Decision Algorithm for Intraprediction in H.264/AVC Video Coding. *IEEE Trans. Circuit and Systems for Video Technology* 15(7), 813–822 (2005)
8. Li, H., Li, Z.G., Wen, C., Chau, L.P.: Fast mode decision for spatial scalability vidoe coding. In: *Proc. of IEEE International Symposium on circuits and Systems*, pp. 3005–3008 (May 2006)
9. Lim, S.H., Yang, J.Y., Jeon, B.W.: Fast coding mode decision for scalable video coding. In: *Proc. of IEEE International Conference on Advanced Communication*, pp. 1897–1900 (February 2008)
10. Li, H., Li, Z.G., Wen, C.: Fast mode decision algorithm for inter-frame coding in fully scalable video coding. *IEEE Trans. Circuit Syst. Video Technol.* 16, 889–895 (2006)
11. Lin, H.C., Peng, H.W., Hang, M.H., Ho, W.J.: Layer adaptive mode decision and motion search for scalable video coding wiht combined coarse gnaular scalability and temporal scalability. In: *Proc. of IEEE International Conference on Image Processing*, pp. 289–292 (September 2007)
12. Bjontegaard, G.: Claculation of average PSNR differences between RD-curves. In: *VCEG-M33, 13th meeting, Austin* (April 2001)

A Novel Rate Control Method for H.264/AVC Based on Frame Complexity and Importance

Haibing Chen¹, Mei Yu^{1,2}, Feng Shao¹, Zongju Peng¹, Fucui Li¹,
and Gangyi Jiang^{1,2}

¹ Faculty of Information Science and Engineering, Ningbo University,
Ningbo-city, 315211, China

² National Key Lab of Software New Technology, Nanjinging University,
Nanjing-city, 210093, China
jianggangyi@126.com

Abstract. In this paper, we present a new rate-control algorithm based on frame complexity and importance (CI) for H.264/AVC video coding. The proposed method aims at selecting accurate quantization parameters for inter-coded frames according to the target bit rates, by accurately predicting frame CI using the statistics of previously encoded frames. Bit budget is allocated to each frame adaptively updated according to its CI, combined with the buffer status. We compare the proposed method with JVT-G012 used by H.264/AVC with the software JM10.1. The PSNR performance of video coding is improved by the proposed method from 0.142 to 0.953 dB, and the BDPSNR performance is improved from 0.248 to 0.541dB. The proposed method can also provide more consistent visual quality and alleviated sharp drops for frames caused by high motions or scene changes with the PSNR standard deviation decreases from 0.134 to 1.514dB.

1 Introduction

The remarkable evolution of video coding technology has underlined the development of a multitude of novel signal compression techniques that aimed to optimise the compression efficiency and quality of service of standard video coders under certain bandwidth [1]. Rate control plays a critical role in the video encoder, although it does not belong to the normative part in video coding standards. It regulates the coded bit stream to satisfy certain given conditions, on the one hand, and enhances the quality of coded video, on the other hand. Some efficient rate control schemes have been proposed and used, such as TM5 for MPEG-2 [2], TMN8 for H.263 [3], VM8 for MPEG-4 [4] and JVT-G012 for H.264/AVC [5].

JVT-G012 uses a fluid flow traffic model to compute the target bit for the current encoding frame and a linear model to predict mean absolute difference (MAD) to solve the chicken and egg dilemma. Lee et al. presented a complexity-based intra-frame rate control algorithm [6], by predicting a relative complexity of a current macroblock (MB) from complexities of its spatially/temporally neighboring MBs. Jing et al. presented an effect of I frame RC by using gradient-based image complexity and exponential R-Qstep model [7]. Zhu et al. used temporal average MAD to replace traditional MAD linear prediction model, increasing the average luminance

PSNR of reconstructed video by up to 0.58 dB [8]. Tu et al. modeled a more accurate rate and distortion functions[9]. The newest scalable video coding specification H.264/SVC, its reference software Joint Scalable Video Model (JSVM) also adopts a JVT-G012-like rate control scheme for its base layer [10]. Yin et al. proposed an optimum bit allocation scheme to improve the rate control accuracy [11], though its complexity factor, simply determined by the encoding frame and its previous one frame, failed to represent the frame complexities over a GOP. It also little consider the different importance of each P frame in a GOP. Many MBs in the subsequent frame after scene change may need to be encoded in intra-mode and need more bits or else it may cause a serious degradation in picture quality.

In this paper, we first define a reasonable factor to describe frame complexity and importance (CI). Then, according to the CI of each frame, an adaptive allocation target bits and buffer strategy among different frames is presented to improve the quality of frames especially for high motions or scene changes. The organization of the paper is as follows. Section 2 briefly introduces preliminary knowledge for later section. In Section 3, a CI-based rate-control method is proposed. For demonstrating the effectiveness of the proposed scheme, and the experimental results are provided in Section 4. Section 5 concludes the paper.

2 Analysis of Frame Layer Rate Control in JVT-G012

In JVT-G012, QPs of I frame and the first P frame in a group-of-pictures (GOP) are calculated based on available channel bandwidth and GOP length. All the remaining forward predicted pictures (P frames) are calculated based on a target bit for each frame and RDO process for the current frame. All bi-directional predicted pictures (B frames) are obtained through a linear interpolation method according to QP of P frames. It is quite important to accurately estimate target bits for the current P frame. In this section, we will review the method used for estimating the target bits in JVT-G012 and analyse the limitation of the existing method.

A fluid traffic model based on the linear tracking theory is employed to estimate target bits for the current P frame [5]. For simplicity, assume a GOP is encoded with IPPP prediction structure. Let N denote total number of frames in a GOP, n_j is the j th frame in a GOP, $u(n_j)$ denote available channel bandwidth, $T_r(n_j)$ be the number of remaining bits before encoding the current frame, $B_c(n_j)$ denote the occupancy of virtual buffer after coding current frame and $A(n_j)$ be the actual of bits generated after encoding a frame. To estimate target bits for the current P frame the fluid traffic model is used to update T_r frame by frame as follows

$$T_r(n_j) = T_r(n_{j-1}) + \frac{u(n_j) - u(n_{j-1})}{F_r} (N - j) - A(n_{j-1}), \quad (1)$$

where $T_r(n_{j-1})$ be the number of remaining bits after encoding last frame. Meanwhile, the target buffer level Tbl for each frame is updated frame by frame as follows

$$Delt_p = \frac{Tbl(n_2) - B_s/8}{N_p - 1}, \quad Tbl(n_j) = Tbl(n_{j-1}) - Delt_p - \frac{u(n_j)}{F_r}. \quad (2)$$

Then linear tracking theory is employed to determine the target bits allocated for the j th frame as follows

$$T_{buf}(n_j) = \frac{u(n_j)}{F_r} + \gamma(Tbl(n_j) - B_c(n_j)), \quad (3)$$

where γ is a constant and its typical value is 0.75 [5]. Meanwhile, the remaining bits are computed by

$$T_{ref}(n_j) = \frac{T_r(n_j)}{R_{PN}(j-1)}, \quad (4)$$

where $R_{PN}(j-1)$ is the number of P frames remaining for encoding. The final target bit R for the j th frame is calculated by

$$R(n_j) = \beta * T_{ref}(n_j) + (1-\beta) * T_{buf}(n_j), \quad (5)$$

where β is a weighting factor and set typically as 0.5 [5].

In Eq.(2), all frames have an equal number of target buffer level. In Eq.(4), the remaining bits T_r is also allocated to all non-coded frames equally. Thus, a buffer nearly full will allocate less target bits to a new frame while a nearly empty buffer will allocate more bits, which will lead to a much smaller quantization parameter regardless of the complexity of frame content. Inaccurately estimate target bits for the current P frame results in fluctuations in picture quality and decrease in coding efficiency.

In the proposed scheme, we focused on T_r and Tbl , that is, the remaining bits and buffer level should be un-equally distributed to all non-coded frames according to frame complexities and importance in the target bit estimation step. In other words, different complex and important frame will get different buffer and bandwidth resource. Details of the improvements will be discussed in Section 3.

3 The Proposed Rate-Control Method Using Frame CI

The basic idea in this paper is to allocate more bits for scene change frames or high complexity frames or for important frames, and less bits for low complexity frames or unimportance frames to achieve constant quality. It is well known that MAD can be a good indication of encoding complexity of the residual component. In the quadratic rate-quantization (R-Q) model, the encoding complexity is usually substituted by MAD [5]. Lee et al. measure 4x4 Intra-block complexity by using MAD with 5x5 statistical window [12]. Based on their contribution, we defined a new factor to describe P frame parameter complexity and importance, denoted as CI, and proposed a rate-control method using frame CI.

3.1 Complexity and Importance Measure of P Frame

Average actual MAD (AMAD) of all previously encoded P frames in GOP is defined to represent the complexity of encoded P frames. AMAD is calculated as follows

$$AMAD(j) = \frac{1}{j} \sum_{k=1}^{j-1} MAD(k), \quad (6)$$

Then, a linear prediction model, like in [5], is employed to calculate the predicted MAD (PMAD) of the current frame by $PMAD(j) = a_1 \times AMAD(j-1) + a_2$. A method similar as updating parameters of R-D model, like in MPEG-4[4], is given to update a_1 and a_2 as

$$a_1 = \frac{j \sum_{k=1}^j MAD(k-1) \times MAD(k) - \sum_{k=1}^j MAD(k-1) \sum_{k=1}^j MAD(k)}{j \sum_{k=1}^j (MAD(k-1))^2 - \left(\sum_{k=1}^j MAD(k-1) \right)^2} \quad (7)$$

$$a_2 = \frac{\sum_{k=1}^j (MAD(k-1))^2 \times MAD(k) - \sum_{k=1}^j MAD(k-1) \sum_{k=1}^j MAD(k) MAD(k-1)}{j \sum_{k=1}^j (MAD(k-1))^2 - \left(\sum_{k=1}^j MAD(k-1) \right)^2}$$

where j is the number of the encoded frames. Relative complexity of encoding frame (RMAD) can be represented as the ratio of the predicted MAD of PMAD and AMAD, and computed by

$$RMAD(j) = \frac{PMAD(j)}{AMAD(j)}. \quad (8)$$

$RMAD$ is a simple and accurate measure of frame complexity, and provides a mechanism to control the target bits estimation. We quantize $RMAD$ with a non-linear strategy, $C(j)$, as follows

$$C(j) = \begin{cases} 0.5 & RMAD \leq 0.8 \\ 0.6RMAD & 0.8 < RMAD \leq 1.0 \\ 0.7RMAD & 1.0 < RMAD \leq 1.8 \\ 1.8 & RMAD > 1.8 \end{cases} \quad (9)$$

Meanwhile, the importance of frame should be considered. Just as in JVT-G012 [5], it deemed that P frame is more important than B frame. It should allocate more bits to P frame. Similarly, a latter P frame is predicted from the former P frames and frames in a GOP may have similar content. Hence, the higher quality the referenced frames are, the smaller different between the referenced frames and the predicted frame will probably be. Thus a video can get a higher quality at same cost of bandwidth. It can be deemed that the distance of each P frame from the initial I frame in a GOP should be considered when allocating bit. Parameter $I(j)$ that denotes the importance of frame is given as

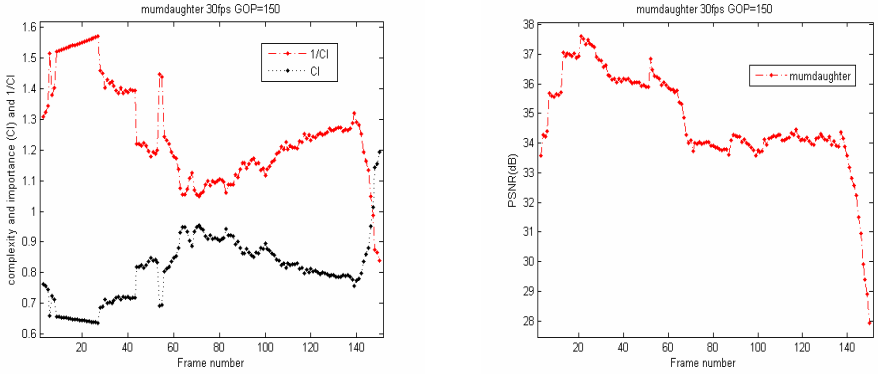
$$I(j) = \frac{R_{PN}(j-1)}{N_p} \quad (10)$$

where $R_{PN}(j-1)$ is the number of P frames remaining for encoding, N_p is total P frame in a GOP. From these analyses, a new parameter $CI(j)$ is defined as frame's complexity and importance (CI), and calculated by

$$CI(j) = C(j) + \zeta I(j) \quad (11)$$

where ζ is a constant, ranging from 1/9 to 1. The parameter $CI(j)$ provides a new measure for global encoding complexity. The equally distributed of the remaining bits and buffer level to all non-coded frames lead to fluctuations in picture quality and decrease in coding efficiency.

Fig. 1(a) shows CI with the junction of scene changes. It is easy to find that the inverses of CI are consistent with the PSNR curve.



(a) CI and 1/CI of ‘Mother and Daughter’ sequence (b) PSNR results for ‘Mother and Daughter’ sequence

Fig. 1. CI, 1/ CI and PSNR for ‘Mother and Daughter’ sequence ($\zeta=1/6$)

3.2 Improved Buffer Allocation Scheme

CI is a simple and accurate measure of frame complexity and importance. Therefore, it can provide a mechanism to control estimation of the target bit. If the frames’ CIs are large, it should allocate more remaining bits and buffer resource to them. From Eq.(4), it can be concluded that $T_{ref}(n_j)$ is directly with CI. Meanwhile, a frame with large CI should take more buffer resource. From Eq.(3), if we want to allocate more buffer resource to a frame, should enlarge $Tbl(n_j)$. From Eq.(2b), if we want to large $Tbl(n_j)$, we should reduce D_{deltip} . For computational simplicity, the improved buffer allocation scheme is presented to adjust the allocation of remaining bits and buffer resource by

$$T_{ref}(n_j) = CI \frac{T_r(n_{i,j})}{R_{pN}(j-1)} \tag{12}$$

$$D_{deltip} = \frac{Tbl(n_2) - B_s/8}{N_p - 1} \times \frac{\sigma}{CI} \tag{13}$$

where σ is a constant range from 0.4 to 0.6. It is noted that the parameters used in the above function all come from empirical experiments with different resolution and-frame rate. The objective of this improvement is to save bits from those frames with relatively less complexity or less importance and allocate more bits to frames with higher complexity or more importance. The final target bits R for the new P-frame can be calculated using equation (5), where β increases to 0.55 from 0.5 so that T_{ref} has more weight than T_{buf} .

4 Experimental Results and Analyses

To evaluate performances of the proposed method, rate control experiments are implemented on QCIF video sequences with different activity and motion features.

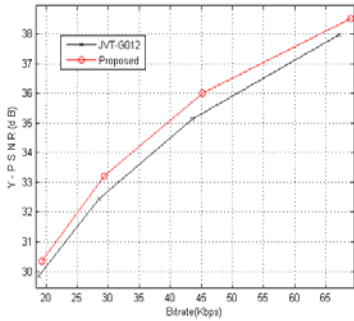
Table 1. Performance comparison between JVT-G012 and the proposed method

Seq.	Tar. (Kbps)	Actual. (Kbps)		Rateerr. (%)		PSNR		
		G012	Pro.	G012	Pro.	G012	Pro.	Imp.
moniter	66.836	67.121	69.011	0.426	3.254	37.969	38.510	0.541
	43.131	43.492	45.211	0.837	4.823	35.133	35.995	0.862
	27.972	28.481	29.305	1.820	4.765	32.441	33.217	0.776
	18.275	18.832	19.318	3.048	5.707	29.863	30.359	0.496
salesman	76.443	76.971	80.059	0.691	4.730	36.520	36.882	0.362
	44.808	45.425	47.556	1.377	6.133	33.143	33.581	0.438
	26.380	26.862	28.044	1.827	6.308	30.518	31.095	0.577
	14.851	15.497	15.929	4.335	7.259	28.341	28.627	0.286
mum-daughter	56.198	56.814	59.139	1.096	5.233	37.963	38.301	0.338
	31.777	32.206	33.371	1.350	5.016	34.423	35.376	0.953
	18.300	18.928	19.268	3.432	5.290	32.164	32.688	0.524
	10.848	11.520	11.553	6.195	6.499	30.168	30.310	0.142
akiyo	38.598	39.300	40.694	1.819	5.430	38.823	39.378	0.555
	24.244	24.940	25.420	2.871	4.851	35.760	36.417	0.657
	15.827	16.451	16.699	3.943	5.510	33.074	33.385	0.311
	11.020	11.513	11.867	4.474	7.686	30.937	31.200	0.263

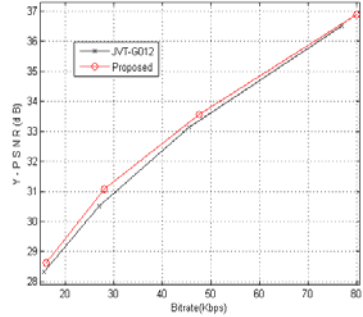
Table 2. PSNR standard deviation, $PSNR\sigma$, comparison between the proposed method and JVT-G012

Sequence	Tar. (Kbps)	$PSNR\sigma$		
		JVT-G012	Proposed	Diff
moniter	66.836	2.567	1.143	1.424
	43.131	3.415	1.901	1.514
	27.972	3.155	2.162	0.993
	18.275	2.399	2.040	0.359
salesman	76.443	2.154	1.89	0.264
	44.808	2.763	2.503	0.260
	26.380	2.622	2.138	0.484
	14.851	1.647	1.535	0.112
mum daughter	56.198	2.838	2.314	0.524
	31.777	3.853	2.528	1.325
	18.300	3.138	2.618	0.520
	10.848	2.448	2.310	0.138
akiyo	38.598	2.353	1.715	0.638
	24.244	2.742	1.967	0.775
	15.827	2.438	2.172	0.266
	11.020	1.725	1.591	0.134

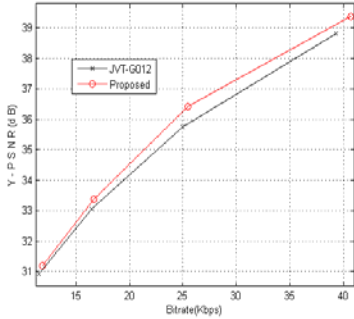
Each test sequence is encoded by IPPP prediction structure with the GOP length of 30. RDO is enabled both in mode decision and motion estimation. All test sequences used are in 4:2:0 format, 150 Frames, ζ in Eq.(11) is set 0.5, σ in Eq.(13) is set 0.5. The test platform is JM10.1 [13]. Like JVT-G012, the bit rates, which are generated by encoding the test sequences with the fixed QPs of 28, 32, 36, and 40, are the target bit rates for an encoder with the proposed rate control scheme. Table 1 illustrates performance comparison between JVT-G012 and the proposed method. Experiments have been carried out on PC with the Intel Core 3.0GHz CPU and 3.25GB RAM.



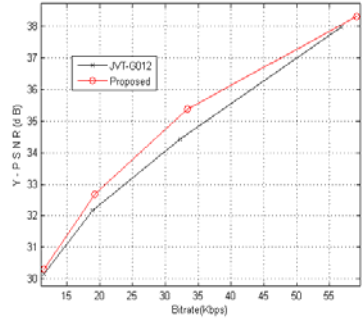
(a) RD performance for moniter
BDPSNR=0.541 dB



(b) RD performance for saleman
BDPSNR=0.248 dB



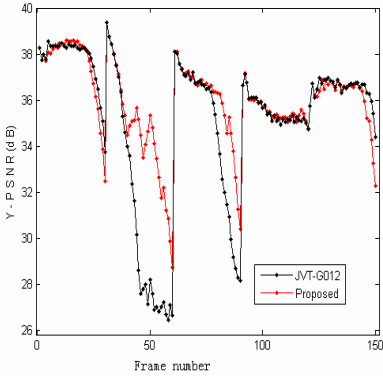
(c) RD performance for akiyo
BDPSNR=0.361 dB



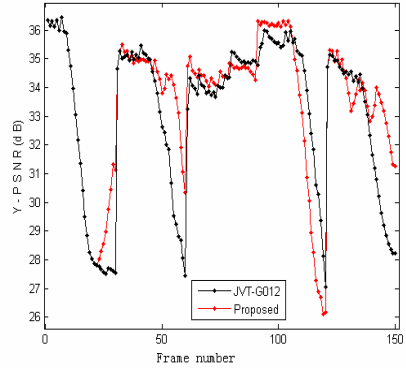
(d) RD performance for mumdaughter
BDPSNR=0.490 dB

Fig. 2. R-D curve comparison between the proposed method and JVT-G012

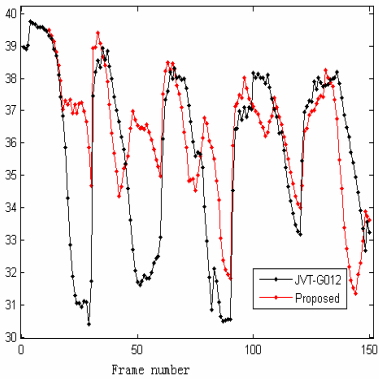
In the table 1, Rate error=(Actual Rate-Target Rate)/ Target Rate. It is clear that JVT-G012 has a rate error range from 0.426% to 6.195%, while the proposed method ranges from 3.254% to 7.686%, the proposed method outperforms JVT-G012. They have similar rate error while the PSNR of the proposed method is improved up to 0.953 dB compared with JVT-G012, and the minimum improvement is 0.142dB. In Table 1, it is seen that the proposed method achieves higher PSNR with negligible and increments in bit rate. In order to compare coding efficiency, rate-distortion



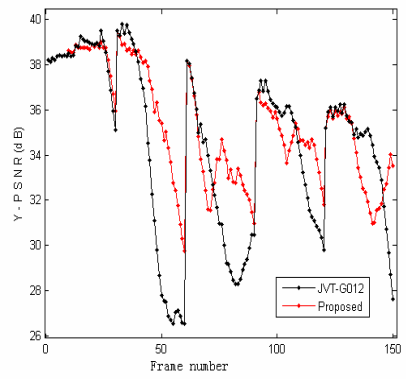
(a) PSNR fluctuation of monitor



(b) PSNR fluctuation of salesman



(c) PSNR fluctuation of akiyo



(d) PSNR fluctuation of mumdaughter

Fig. 3. PSNR fluctuation comparison between the proposed method and JVT-G012

curve is given in Fig. 2, with Bjontegaard delta PSNR (BDPSNR) [14] comparing the difference between two RD curves.

From Fig.2, it is clear that the proposed method has a better coding efficiency, especially in middle bandwidth. The BDPSNR of the proposed method is improved by up to 0.541 dB for monitor, the minimum improvement is 0.248dB for salesman, a sequence with normal motion and less scene change sequence. Additional comparisons are given in Table 2, where PSNR standard deviation,

$$PSNR\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (PSNR_i - \overline{PSNR})^2}$$

, is used to describe PSNR fluctuation. Standard

deviation is a parameter to measure numerical value spread out degree in mathematics. In Table 2, it is clear that the PSNR standard deviation, $PSNR\sigma$, of the proposed method decreases from 0.134dB to 1.514dB compared to JVT-G012, it implies that the rate-control accuracy of the proposed method is better that of JVT-G012. Clearer results are listed in figure show the PSNR fluctuation under the condition as target bit

rates for an encoder are generated by coding the test sequences with the fixed QP of 32. Other condition is the same as the front experiments.

Fig. 3. illustrates that the proposed method can avoid drastic visual quality variation caused by scenes. Smaller PSNR fluctuation implies more stable visual quality which is highly desired in video coding.

5 Conclusion

In this paper we have presented a new rate control technique to improve the rate control in H.264/AVC video coding. The proposed method considers scene's characteristics and its bit allocation is more reasonable so that it can maintain a video stream with a smoother PSNR variation which is highly desirable in real-time video coding and transmission. Meanwhile, experimental results show that the coding efficiency has been improved in the proposed method.

In future work, the proposed improvements will be extended to rate control in stereo video communication. Bitrate budget is first allocated to each stereo image frame adaptively updated according to bandwidth and buffer status, combined with complexity and importance of stereo image frame. Then it should also allocate bits inter views according to a parameter of frame complexity and importance to keep the quality of each view totally equal. And it may be helpful to get a more consistent visual quality when scene switching occurs in stereo video.

Acknowledgement

This work was supported by Natural Science Foundation of China (60872094, 60832003), the projects of Chinese Ministry of Education (200816460003), the Scientific Research Fund of Zhejiang Provincial Education Department (Z200909361), and the National 863 Project of China (2009AA01Z327).

References

1. Chen, Z., Ngan, K.N.: Recent advances in rate control for video coding. *Signal Processing: Image Communication* 22, 19–38 (2007)
2. MPEG-2 Test Model 5, Doc. ISO/IEC JTC1/SC29 WG11/93-400 (April 1993)
3. Corbera, J.R., Lei, S.: Rate control in DCT video coding for low delay communication. *IEEE Trans. on CSVT* 9, 172–185 (1999)
4. Lee, H.J., Chiang, T.H., Zhang, Y.Q.: Scalable rate control for MPEG-4 video. *IEEE Trans. on CSVT* 10, 878–894 (2000)
5. Li, Z.G., Pan, F., Lim, K.P., et al.: Adaptive basic unit layer rate control for JVT. In: *The 7th JVT Meeting, JVT-G012-rl, Thailand (March 2003)*
6. Lee, Y.G., Song, B.C.: An Intra-Frame Rate Control Algorithm for Ultralow Delay H.264/Advanced Video Coding (AVC). *IEEE Trans. on CSVT* 19, 747–752 (2009)
7. Jing, X., Chau, L., Siu, W.: Frame complexity-based rate quantization model for H.264/AVC intra- frame rate control. *IEEE Signal Processing Letters* 15, 373–376 (2008)

8. Zhu, T., Zhang, X.: A Novel Rate Control Scheme for H.264/SVC Base Layer. In: Zhu, T., Zhang, X. (eds.) International Conference on Wireless Communications & Signal Processing, WCSP 2009, Nanjing, China (November 2009)
9. Tu, Y., Yang, J., Sun, M.: Rate-distortion modeling for efficient H.264/AVC encoding. *IEEE Trans. on CSVT* 17, 530–543 (2007)
10. Reichel, J., Schwarz, H., Wien, M.: Joint Scalable Video Model JSVM-12 text, Joint Video Team, Doc. JVT-Y202, 10 (2007)
11. Yin, M., Wang, H.: A rate control scheme for H.264 video under low bandwidth channel. *J. Zhejiang Univ SCIENCE A* 7, 990–995 (2006)
12. Lee, G., Lin, H., Wang, M.: Rate control algorithm based on intra-picture complexity for H.264/AVC. *IET Image Processing* 3, 26–39 (2009)
13. JM Reference Software Version 10.1., <http://iphone.hhi.de/suehring/tml/download/>
14. Bjontegaard, G.: Calculation of Average PSNR Differences between RD-curves, ITU-T SC16/Q6. In: 13th VCEG Meeting, Austin, Texas, USA (April 2001)

Digital Image Tamper Detection Based on Multimodal Fusion of Residue Features

Girija Chetty¹, Julian Goodwin², and Monica Singh²

¹ Faculty of Information Sciences and Engineering,
University of Canberra, Australia
girija.chetty@canberra.edu.au

² Video Analytics Pty. Ltd. Melbourne, Australia

Abstract. In this paper, we propose a novel formulation involving fusion of noise and quantization residue features for detecting tampering or forgery in video sequences. We reiterate the importance of feature selection techniques in conjunction with fusion to enhance the tamper detection accuracy. We examine three different feature selection techniques, the independent component analysis (ICA), fisher linear discriminant analysis (FLD) and canonical correlation analysis (CCA) for achieving a more discriminate subspace for extracting tamper signatures from quantization and noise residue features. The evaluation of proposed residue features, the feature selection techniques and their subsequent fusion for copy-move tampering emulated on low bandwidth Internet video sequences, show a significant improvement in tamper detection accuracy with fusion formulation.

Keywords: image tampering, digital forensics, feature selection, image fusion.

1 Introduction

Digital Image tampering or forgery has become major problem lately, due to ease of artificially synthesizing photographic fakes- for promoting a story by media channels and social networking websites. This is due to significant advances in computer graphics and animation technologies, and availability of low cost off-the-shelf digital image manipulation and cloning tools. With lack of proper regulatory frameworks and infrastructure for prosecution of such evolving cyber-crimes, there is an increasing dissatisfaction about increasing use of such tools for law enforcement, and a feeling of cynicism and mistrust among the civilian operating environments.

Another problem this has lead to, is a slow diffusion of otherwise extremely efficient image based surveillance and identity authentication technologies in real-world civilian operating scenarios. In this paper we propose a novel algorithmic framework for detecting image tampering and forgery based on extracting noise and quantization residue features, their transformation in cross-modal subspace and their multimodal fusion for intra-frame and inter-frame image pixel sub blocks in video sequences. The proposed algorithmic models allow detecting the tamper or forgery in low-bandwidth video (Internet streaming videos), using blind and passive tamper detection techniques and attempt to model the source signatures embedded in camera

pre-processing chain. By sliding segmentation of image frames, we extract intra-frame and inter-frame pixel sub-block residue features, transform them into optimal cross-modal subspace, and perform multimodal fusion to detect evolving image tampering attacks, such as JPEG double compression, re-sampling and retouching. The promising results presented here can result in the development of digital image forensic tools, which can help investigate and solve evolving cyber crimes.

2 Background

Digital image tamper detection can use either active tamper detection techniques or passive tamper detection techniques. A significant body of work, however, is available on active tamper detection techniques, which involves embedding a digital watermark into the images when the images are captured. The problem with active tamper detection techniques is that, not all camera manufacturers embed the watermarks, and in general, most of the customers have a dislike towards cameras which embed watermarks due to compromise in the image quality. So there is a need for passive and blind tamper detection techniques with no watermark available in the images.

Passive and blind image tamper detection is a relatively new area and recently some methods have been proposed in this area. Mainly these are of two categories [1, 2, 3, 4]. Fridrich [4] proposed a method based on hardware aspects, using the feature extracted from photos. This feature called sensor pattern noise is due to the hardware defects in cameras, and the tamper detection technique using this method resulted in an accuracy of 83% accuracy. Chang [5] proposed a method based on camera response function (CRF), resulting in detection accuracy of 87%, at a false acceptance rate (FAR) of 15.58%. Chen et al. [6] proposed an approach for image tamper detection based on a natural image model, effective in detecting the change of correlation between image pixels, achieving an accuracy of 82%. Gou et al [7] introduced a new set of higher order statistical features to determine if a digital image has been tampered, and reported an accuracy of 71.48%. Ng and Chang [8] proposed bi-coherence features for detecting image splicing. This method works by detecting the presence of abrupt discontinuities of the features and obtains an accuracy of 80%. Popescu and Farid [3] proposed different CFA (colour filter array) interpolation algorithms within an image, reporting an accuracy of 95.71% when using a 5x5 interpolation kernel for two different cameras. A more complex type of passive tamper detection technique, known as “copy-move tampering” was investigated by Bayram, Sencar, Dink and Memon [1,2] by using low cost digital media editing tools such as Cloning in Photoshop. This technique usually involves covering an unwanted scene in the image, by copying another scene from the same image, and pasting it onto the unwanted region. Further, the tamperer can use retouching tools, add noise, or compress the resulting image to make it look genuine and authentic. Finally, detecting tampers based on example-based texture synthesis scheme was proposed by Criminisi et al[9] that is based on filling in a region from sample textures. It is one of the state-of-the-art image inpainting or tampering schemes. Gopi et al in [10] proposed a pattern recognition formulation and used auto regression coefficients and neural network classifier for tamper detection.

One of the objectives of the work reported here is development of robust and automatic tamper detection framework for low bandwidth Internet streamed videos where most of the fingerprints left by tamperer can be perturbed by heavy compression. However, by fusing multiple image tampering detectors, it could be possible to uncover the tampering in spite of the heavy compression, as different detectors use cues and artifacts at different stages of the image formation process. So if an image lacks certain cues, a complementary detector would be used for making a decision. For example, a copy move forgery might have been created with two source images of similar quantization settings but very different cameras. In this case, the copy move forgery can be successfully detected by a different detector. We thus benefit from having several tamper detection modules at hand rather than only using the one type of detector. Another advantage of fusing several detector outputs to make a final decision is that, if one of the detector outputs noisy and erroneous scores, the other detectors could complement and enhance the reliability of the tamper decision. Therefore, the advantage of fusion is twofold: to handle images which were subjected to multiple, diverse types of tampering, and to boost the detection robustness and accuracy by making different modules work with each other. The challenge, however, lies in the synergistic fusion of diverse detectors as different detectors are based on different physical principles and segmentation structures.

We formulate the fusion problem in a Bayesian pattern recognition framework and use well known Gaussian Mixture Models for the task. The approach is based on detecting the tamper from the multiple image frames, by extracting noise and quantization residue features in intra-frame and inter-frame pixel sub blocks (we refer to pixel sub blocks hence forth in this paper as macro blocks), transforming them into optimal feature subspace (ICA, CCA or FLD) to extract the maximal correlation properties, and use GMM classifier to establish possible tampering of video. To enhance the confidence level of one of the tamper detector, we either perform a fusion of detector scores (late fusion) or fuse the features first and perform the classification later (feature fusion). The approach extends the noise residue features reported by Hsu et al in [11] and expands the pattern recognition formulation proposed by Gopi et al in [10]. The approach is blind and passive, based on the hypothesis, that typical tampering attacks such as double compression, re-sampling and retouching can inevitably disturb the correlation properties of the macro blocks within a frame (intra-frame) as well as between the frames (inter-frame) and can distinguish the fingerprints or signatures of genuine video from tampered video frames. The rest of the paper is organized as follows. Next Section describes the formulation of fusion problem. The details of the experimental results for the proposed fusion scheme is described in Section 4. The paper concludes in Section 5 with some conclusions and plan for further work.

3 Formulating the Fusion Problem

The processing pipeline once the images or video is captured consists of several stages. First, the camera sensor (CCD) captures the natural light passing through the optical system. Generally, in consumer digital cameras, every pixel is detected by a CCD detector, and then passed through different colour filters called Color Filter

Array (CFA). Then, the missing pixels in each color planes are filled in by a CFA interpolation. Finally, operations such as demosaicing, enhancement and gamma correction are applied by the camera, and converted to a user-defined format, such as RAW, TIFF, and JPEG, and stored in the memory.

Since the knowledge about the source and exact processing (details of the camera) used is not available for application scenarios considered in this work (low-bandwidth Internet video sequences), and which may not be authentic and already tampered, we extract a set of residual features for macro blocks within the frame and between adjacent frames from the video sequences. These residual features try to model and extract the fingerprints for source level post processing within any camera, such as denoising, quantization, interlacing, de-interlacing, compression, contrast enhancement, white balancing, image sharpening etc. In this work, we use only two types of residual features: noise residue features and quantization residue features.

The noise and quantization residue features were first extracted from 32 x 32 pixel intra-frame and inter-frame macro blocks of the video sequences. The details of noise and quantization residue features are described in [3], [4] and [11]. A feature selection algorithm was used to select those features that exhibit maximal significance. We used feature selection techniques based on three different techniques: Fisher linear discriminant analysis (FLD), canonical correlation Analysis (CCA), and Independent component analysis (ICA). The details of the three feature selection techniques is described in [12], [13].

4 Experimental Results

The video sequence data base from Internet movie sequences was collected and partitioned into separate subsets based on different actions and genres. The data collection protocol used was similar to the one described in [14]. Figure 1 shows screenshots corresponding to different actions, along with emulation of copy move tampered scenes and the detection of tampered regions with the proposed approach.

Different sets of experiments were conducted to evaluate the performance of the proposed feature selection approaches, namely, the ICA, the FLD and the CCA and their fusion (late fusion or feature fusion) in terms of tamper detection accuracy. The experiments involved a training phase and a test phase. In the training phase, a Gaussian Mixture Model for each video sequence from data base was constructed [15]. In the test phase, copy-move tamper attack was emulated by artificially tampering the training data. The tamper processing involved copy cut pastes of small regions in the images and hard to view affine artefacts. Two different types of tampers were examined. An intra-frame tamper, where the tampering occurs in some of the macro blocks within the same frame, and inter-frame tamper, where macro blocks from adjacent frames were used. However, in this paper, we present and discuss results for the intra-frame tamper scenario only.

As can be seen from Table 1, which show the tamper detection results in terms of % accuracy, the performance of noise residue and quantization residue features without feature selection, the improvement achieved by using feature selection techniques, and the robustness achieved by fusing the sub space features (feature level fusion) or the scores. We compared the performance of proposed feature selection and fusion techniques with feature selection based on autoregressive coefficients and neural network classification proposed by Gopi et al in [10].



Fig. 1. Row 1: Screenshots from Internet streamed video sequences; Row 2: Copy-move tamper emulation for the scene; Row 3: Detection of tampered regions in the scene

As can be seen in Table 1, the single mode noise residue features perform better than quantization residue features. For both noise residue and quantization residue features, the CCA, ICA and FLD features perform better than ARC features. CCA features result in better accuracy for noise residue features as compared to others, as they are based on canonical correlation analysis that can extract maximal correlation properties better than features based on Fisher linear discriminant analysis. However, for quantization residue features, the ICA features perform better than CCA features showing that quantization information perturbed by tampering may not be necessarily correlated, but could contain certain independent components. By fusing intra-frame and inter-frame macro block features, we can see a better performance is achieved. This shows that better correlation information can be extracted when multiple frames are used for detecting tampers. Further, by fusing the two detectors, the detectors based on noise residue features and quantization residue features, we can see that a better performance is achieved as the two detectors complement each other, resulting in a consistent and stable performance. This can be expected as quantization artefacts for low-bandwidth video can significant damage tamper related correlation properties. However, by using a hybrid fusion of quantization and noise residue features from macro blocks, and using different feature selection techniques, we can see that a better performance is achieved.

As we are using a pattern recognition formulation, the classifier used for making decision on tampering is also equally important (in addition to an appropriate feature selection technique). Hence the next experiment involved examining the performance of GMM classifier with the neural network (NN) classifier based on back propagation network proposed in [10], and the support vector machine (SVM) classifier based on RBF kernel proposed in [17]. The results from this experiment are shown in Table 2 and Table 3. Since the experiments reported in Table 1 resulted in CCA and ICA features as the best performing features, we used CCA and ICA features for experimental results shown in Table 2 and Table 3.

Table 1. Evaluation of noise and quantization residue features for emulated copy-move tamper attack (% Accuracy); $\tilde{f}_{Intra-Inter}$ (noise residue features); $f_{Intra-Inter}$ (quantization residue features)

Internet movie data subset	% Accuracy			
Different Residue features and their fusion	CCA	ICA	FLD	ARC[10]
f_{Intra} (Intra-frame noise residue features)	83.2	83.4	83.6	80.2
f_{Inter} (Inter-frame noise residue features)	83.8	83.1	83.4	83.1
\tilde{f}_{Intra} (Intra-frame quant. residue features)	77.28	80.26	76.23	74.33
\tilde{f}_{Inter} (Inter-frame quant. residue features)	72.65	78.27	71.44	69.45
$f_{Intra-Inter}$ (feature fusion- noise residue)	86.6	86.1	85.27	83.78
$\tilde{f}_{Intra-Inter}$ (feature fusion- quant residue)	80.55	82.34	79.66	77.22
$f_{Intra-Inter} + \tilde{f}_{Intra-Inter}$ (hybrid fusion)	89.56	88.85	86.22	84.33

Table 2. (% Accuracy) Performance for noise and quantization residue features and their fusion for GMM vs. NN classifier

% Accuracy	GMM Classifier	SVM Classifier	NN Classifier [10]
Different Residue features and their fusion	CCA features	CCA features	CCA features
f_{Intra} (Intra-frame noise residue features)	83.2	83.4	81.4
f_{Inter} (Inter-frame noise residue features)	83.8	83.5	80.6
\tilde{f}_{Intra} (Intra-frame quant. residue features)	77.28	78.18	75.77
\tilde{f}_{Inter} (Inter-frame quant. residue features)	72.65	74.43	70.53
$f_{Intra-Inter}$ (feature fusion- noise residue)	86.6	84.96	83.22
$\tilde{f}_{Intra-Inter}$ (feature fusion- quant residue)	80.55	82.43	77.23
$f_{Intra-Inter} + \tilde{f}_{Intra-Inter}$ (hybrid fusion)	89.56	90.56	83.45

Table 3. (% Accuracy) Performance for noise and quantization residue features and their fusion for GMM vs. NN classifier

% Accuracy	GMM Classifier	SVM Classifier	NN Classifier [10]
Different Residue features and their fusion	ICA features	ICA features	ICA features
f_{Intra} (Intra-frame noise residue features)	83.2	83.5	81.4
f_{Inter} (Inter-frame noise residue features)	83.8	83.7	80.6
\tilde{f}_{Intra} (Intra-frame quant. residue features)	77.28	78.28	75.77
\tilde{f}_{Inter} (Inter-frame quant. residue features)	72.65	73.65	70.53
$f_{Intra-Inter}$ (feature fusion- noise residue)	86.6	85.9	83.22
$\tilde{f}_{Intra-Inter}$ (feature fusion- quant residue)	80.55	81.34	77.23
$f_{Intra-Inter} + \tilde{f}_{Intra-Inter}$ (hybrid fusion)	89.56	91.59	83.45

As can be observed in Table 2 and Table 3, the three classifiers perform differently for different feature selection techniques. For all three feature selection techniques GMM and SVM perform much better than the NN classifier. However, for quantization residue features, the ICA features results in better performance as compared to CCA features, whereas for noise residue features, CCA gives better performance. Further, the SVM classifier performs better than the GMM classifier, for quantization features with ICA feature selection technique. When we perform a fusion two detectors complement each other and resulting in synergistic fusion with combination of ICA and SVM processed quantization features and CCA and GMM processed noise residue features resulting in best performance. The experimental analysis indicates that for detection of image tampering in low bandwidth video sequences, we need to use pattern recognition and fusion based formulation, This formulation allows both linear and nonlinear correlation properties of the tamper scenarios. In this study we have shown for only two simple types of camera image post-processing features. However, other features corresponding to interlacing and de-interlacing artefacts, demosaicing, resampling, touching and blurring need to be examined for characterising the tampering process. This will be the objectives of the future work.

5 Conclusions

In this paper, we investigated a novel approach for video tamper detection in low-bandwidth Internet video sequences using a pattern recognition and information fusion formulation. The approach uses different types of residue features from

intra-frame and inters frame macro blocks, and transforms them into more discriminatory subspace based on different feature selection techniques. We examined ICA, CCA and FLD techniques as three different feature selection techniques for two different image residue features, the noise residue features and quantization residue features.

Further, we propose a fusion of subspace features and examine the performance of fusion formulation with three different types of classifier structures: NN classifier, GMM classifier and SVM classifier. The experimental results show that detection of tamperers in low bandwidth internet video sequences is a challenging task, as traces of tampering (which leaves traces of periodicity and correlation in macro blocks) can be damaged by heavy compression used for reducing the bandwidth. However, by using a pattern recognition and fusion formulation, it is possible to characterise the tamper and use alternate complementary detector. Further work will focus on examining the properties of the image at optical level and detecting the perturbations caused by tampering and extension of propose fusion formulation for development of robust tamper detection tools.

References

- [1] Bayram, S., Sencar, H.T., Memon, N.: An Efficient and Robust Method For Detecting Copy-Move Forgery. In: Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2009, Taiwan (June 2009)
- [2] Dirik, A.E., Memon, N.: Image Tamper Detection Based on Demosaicing Artifacts. In: Proceedings IEEE ICIP Cairo, Cairo Egypt, November 09 (2009)
- [3] Popescu, A.C., Farid, H.: Exposing Digital Forgeries by Detecting Traces of Re-sampling. IEEE Transactions on signal processing 53(2) (February 2005)
- [4] Fridrich J., David, S., Jan, L.: Detection of Copy-Move Forgery in Digital Images, <http://www.ws.binghamton.edu/fridrich/Research/copymove.pdf>
- [5] Hsu, Y.F., Chang, S.F.: Detecting Image Splicing Using Geometry Invariants and Camera Characteristics Consistency. In: ICME, Toronto, Canada (July 2006)
- [6] Shi, Y.Q., Chen, C., Chen, W.: A natural image model approach to splicing detection. In: Proc. ACM Multimedia Security Workshop, Dallas, Texas, pp. 51–62 (September 2007)
- [7] Gou, H., Swaminathan, A., Wu, M.: Noise Features for Image Tampering Detection and Steganalysis. In: Proc. of IEEE Int. Conf. On Image Processing (ICIP 2007), San Antonio, TX (September 2007)
- [8] Ng, T.T., Chang, C.S.F., Lin, Y., Sun, Q.: Passive-blind Image Forensics. In: Zeng, W., Yu, H., Lin, C.Y. (eds.) Multimedia Security Technologies for Digital Rights. Elsevier, Amsterdam (2006)
- [9] Criminisi, A., Perez, P., Toyama, K.: Region filling and object removal by exemplar-based image inpainting. IEEE Trans. Image Process. 13(9), 1200–1212 (2004)
- [10] Gopi, E.S., Lakshmanan, N., Gokul, T., KumaraGanesh, S., Shah, P.R.: Digital Image Forgery Detection using Artificial Neural Network and Auto Regressive Coefficients. In: Proceedings Canadian Conference on Electrical and Computer Engineering, Ottawa, Canada, May 7-10, pp. 194–197 (2006)
- [11] Hsu C., Hung T., Lin C., Hsu C.: Video Forgery Detection Using Correlation of Noise Residues, <http://www.ee.nthu.edu.tw/~cwlin/pub/mmsp08forensics.pdf> (retrieved on 11/3/2010)

- [12] Martinez, A.M., Kak, A.C.: PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23(2), 228–233 (2001), <http://www.ece.osu.edu/~aleix/pami01.pdf>
- [13] Borga, M., Knutsson, H.: Finding Efficient Nonlinear Visual Operators using Canonical Correlation Analysis. In: *Proc. of SSAB 2000*, Halmstad, pp. 13–16 (2000)
- [14] Laptev, I., Marszałek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: *Proc. CVPR 2008*, Anchorage, USA (2008)
- [15] Chetty, G., Wagner, M.: Robust face-voice based speaker identity verification using multilevel fusion. *Image and Vision Computing* 26(9), 1249–1260 (2008)
- [16] Hyvarinen, A., And Oja, E.: Independent Component Analysis: Algorithms and Applications. *Neural Networks* 13(4-5), 411–430 (2000)
- [17] Farid, H., Lyu, S.: Higher-Order Wavelet Statistics and their Application to Digital Forensics. In: *IEEE Workshop on Statistical Analysis in Computer Vision*, Madison, WI (2003)

Fire Detection in Color Images Using Markov Random Fields

David Van Hamme^{1,2}, Peter Veelaert², Wilfried Philips¹, and Kristof Teelen^{1,2}

¹ Ghent University/IBBT (IPI)

² University College Ghent (Vision Systems)

Abstract. Automatic video-based fire detection can greatly reduce fire alert delay in large industrial and commercial sites, at a minimal cost, by using the existing CCTV camera network. Most traditional computer vision methods for fire detection model the temporal dynamics of the flames, in conjunction with simple color filtering. An important drawback of these methods is that their performance degrades at lower framerates, and they cannot be applied to still images, limiting their applicability. Also, real-time operation often requires significant computational resources, which may be unfeasible for large camera networks. This paper presents a novel method for fire detection in static images, based on a Markov Random Field but with a novel potential function. The method detects 99.6% of fires in a large collection of test images, while generating less false positives than a state-of-the-art reference method. Additionally, parameters are easily trained on a 12-image training set with minimal user input.

1 Introduction

Fire detection is an important component of industrial and commercial site surveillance systems with regard to personnel and material safety. Nearly all of the currently employed systems rely on dedicated sensors and manually activated fire alarms. To detect fire as early as possible, a combination of different sensor types is often made, linked by sensor fusion methods to improve reliability. Examples of such techniques include Bao *et al.* [1], who use temperature and photo-electric smoke sensors, and Li *et al.* [2], whose techniques rely on multi-spectral cameras. These systems however, are impractical or too expensive for covering large sites, especially outdoors, due to the required sensor density. A cheap and effective alternative is the use of computer vision-based techniques in conjunction with digital cameras or CCTV networks. The main advantages are the large coverage area offered by a single camera, and the possibility of integration with existing surveillance camera systems.

The state-of-the-art fire detection methods in computer vision typically consist of two main parts, modelling the most characteristic aspects of fire in video. The first aspect is spectral information. All methods employ a color filter of some sort, usually based on a fixed set of rules. The second aspect concerns the temporal dynamics of flames, often combined with spatial characteristics. The spatio-temporal modelling of fire in video was first described by Healey

et al. [3] in 1993, with more recent contributions by Liu *et al.* [4] and Töreyin *et al.* [5]. Other examples of temporal properties of fire used in fire detection include standard background subtraction [6], flame growth and propagation [7,8], intensity and boundary flicker [9], area, roundness and circumference deviation [10], edge dynamics [11] and temporal contour analysis [12].

An important limitation of all these methods is that their performance degrades at lower framerates, as accurate modelling of flame dynamics requires a high temporal resolution, and they cannot be applied to still images. Additionally, the high video data rates combined with the requirement of real-time operation mean that significant computational resources are necessary to monitor a single video stream. This is an important concern for large camera networks. These two drawbacks also inhibit the use of the methods for low-power, wireless camera systems, where frame rates are low to save transmission time and thus save battery power, or where processing is integrated in the camera itself. Some efforts have been made to produce fire detection systems for still images, notably by Noda *et al.* [13], who employed color histogram models for tunnel security monitoring. For use in a more general setting, the static color filters used in the dynamic methods can be used and improved upon, but they still yield a high false alarm rate [14]. Also, as the filters rely on a predefined set of rules, they require time-consuming parameter tuning.

In this paper, a novel method is presented for fire detection on static images. Rather than using a set of rules in color space, the image data is treated as a Markov Random Field (MRF). MRF theory is a powerful tool for modeling contextual dependencies, and has successfully been applied to a variety of texture classification problems [15,16]. The MRF we propose employs a custom potential function shaped by training data. A classifier evaluates the energy function of the MRF per image block to detect blocks on the border of flames. The method is shown to yield near perfect detection rates on a variety of fires, while generating less false alarms than a state-of-the-art fixed-threshold color filter.

2 The MRF Model

Markov Random Fields theory is a branch of probability theory developed for modeling contextual dependencies in physical phenomena. In computer vision, it is primarily used for labeling problems, to establish probabilistic distributions of interacting labels. A thorough description of the application of MRFs to vision problems can be found in the book “MRF Modeling in Computer Vision” by S. Z. Li [17]. The basic principles, terminology and notation are described below.

Let $\mathbf{S} = \{i | i = 1 \dots m\}$ be an index set corresponding to a set of sites in a Euclidian space (e.g. a regular two-dimensional lattice), in which each site is uniquely defined by its index i , and let \mathbf{L} be a discrete or continuous set of labels. Let $\mathbf{F} = \{F_1, \dots, F_m\}$ be a family of random variables defined on \mathbf{S} , in which each random variable F_i takes a value from a label set \mathbf{L} . The label of the random variable F_i will be denoted f_i . Assuming a discrete label set, the probability that F_i takes on a certain label f_i is given by $P(f_i)$. The family \mathbf{F}

is called a random field on \mathbf{S} . The joint probability of the random field taking a particular combination of values is denoted $P(\mathbf{f})$.

A Markov Random Field is defined as a random field in which the probability $P(f_i)$ is only dependent on f_i and some of its neighbors. Therefore, a neighborhood system \mathbf{N} is defined as

$$\mathbf{N} = \{\mathbf{N}_i | i \in \mathbf{S}\} \quad (1)$$

where \mathbf{N}_i is the index set of sites neighboring i . The neighboring relationship has the following properties:

1. a site is not a neighbor of itself: $i \notin \mathbf{N}_i$,
2. the neighboring relationship is mutual: $i \in \mathbf{N}_{i'} \iff i' \in \mathbf{N}_i$.

For a regular lattice \mathbf{S} , the neighboring set of i is usually defined as the set of sites within a radius of i . Note that sites at or near the boundary of the lattice have fewer neighbors. The Markovianity constraint is then expressed by

$$P(f_i | \mathbf{f} - \{f_i\}) = P(f_i | f_{\mathbf{N}_i}) \quad (2)$$

where $\mathbf{f} - \{f_i\}$ denotes all values of the random field except for f_i itself, and $f_{\mathbf{N}_i} = \{f_{i'} | i' \in \mathbf{N}_i\}$ stands for the labels at the sites neighbouring i .

Let us construct a graph on \mathbf{S} in which the edges represent the neighboring relationships. Now consider the cliques in this graph. A clique is a subset of vertices so that every two vertices are connected by an edge. In other words, the cliques represent sites which are all neighbors to each other. Thus, a clique consists of either a single site, or a pair of neighboring sites, or a triple, and so on. The collection of single-site and pair-site cliques will be denoted by \mathbf{C}_1 and \mathbf{C}_2 respectively, where

$$\mathbf{C}_1 = \{\{i\} | i \in \mathbf{S}\} \quad (3)$$

$$\mathbf{C}_2 = \{\{i, i'\} | i' \in \mathbf{N}_i, i \in \mathbf{S}\}. \quad (4)$$

The energy function $U(\mathbf{f})$ is a measure of the likeliness of the occurrence of \mathbf{f} for a given model. For single-site and pair-site cliques, it is defined as

$$U(\mathbf{f}) = \sum_{i \in \mathbf{S}} V_1(f_i) + \sum_{i \in \mathbf{S}} \sum_{i' \in \mathbf{N}_i} V_2(f_i, f_{i'}) \quad (5)$$

where V_1 and V_2 denote potential functions for single-site and pair-site cliques. Lower energy of the joint distribution represents a better fit of the model to the data.

When applied to digital images, the sites correspond to pixel locations, and the neighborhood system is usually either 4-connectedness or 8-connectedness. For a 4-connected system, the four types of pair-site clique that any non-edge pixel belongs to are shown in figure [□](#).

A type of MRF of particular interest to labeling problems in computer vision is the Multi-Level Logistic (MLL) model. In an MLL, the potential functions are defined as

$$V_1(f_i) = \alpha_{f_i} \quad (6)$$

where α_{f_i} is the potential associated with the label f_i , and

$$V_2(f_i, f_{i'}) = \begin{cases} \beta & |f_i = f_{i'} \\ -\beta & |f_i \neq f_{i'} \end{cases} \quad (7)$$

where β is the potential for pair-site cliques. For the 4-connected neighborhood system, each non-edge pixel belongs to four different pair-site cliques, as shown in figure 1. The reason why this model is often used in computer vision, is that for $\beta < 0$, the MLL model acts as a smoothness prior. The potential function then favors smooth distributions with blob-like regions of uniform labels, which is a desirable property in labeling algorithms.

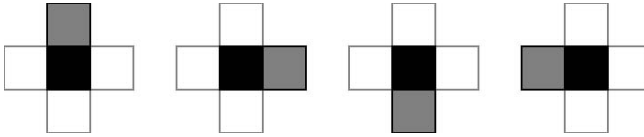


Fig. 1. The four types of pair-site cliques any non-edge pixel belongs to in a 4-connected neighborhood. Black is the pixel concerned, gray is the additional pixel that makes up the clique, outlined in gray is the neighborhood.

For our fire detection application, we want to use the MRF to model the typical red to yellow color texture found in flames. The label set \mathbf{L} therefore consists of discrete color labels, obtained by binning the hue channel in HSV color space. The hue is divided into n evenly spaced bins, with $n \geq 6$, as separation of yellow and red are essential. typically, $n = 8$. Now we must choose the potential functions V_1 and V_2 so that the joint energy $U(\mathbf{f})$ is low for fire areas and high for background. There are two properties of fire we would like to model in the MRF. The first property is the typical color range of the fire pixels. This can be expressed in the potential function for single-site cliques, V_1 , by choosing the values of α_{f_i} low for typical fire colors and high for the others. The second property we want to model is spatial hue variation, reflecting the color gradients typically present in flames. This property will help set apart actual fires from uniformly fire-colored objects, e.g. fire trucks, billboards or fire-colored clothing items. This *unsmoothness* prior can be implemented in the potential function for pairwise cliques, V_2 , by specifying a positive value for the constant β .

A straightforward way to construct a classifier from this model is to evaluate the joint energy $U(\mathbf{f})$ per 4×4 pixel block of the image, and setting a threshold on this energy below which the block is classified as belonging to fire. Our experiments have shown that this technique works and can produce adequate detection rates. However, the false positive rates do not display a significant improvement over state-of-the-art color-based methods [14]. This can be attributed to the simple color space binning. Since V_2 favors any kind of label variation, pair-site cliques consisting of different non-fire colors will also generate low energy. This limits the usefulness of the potential function V_2 and shifts the importance towards V_1 , thereby diminishing the advantage the MLL should theoretically offer.

In practice, this means after optimizing the parameters, β becomes insignificant compared to the values α_{f_i} . A solution to this problem is given in the next section.

3 A Custom Potential Function

One way to resolve the issue described above, would be to adapt the color segmentation to obtain a more intelligent labeling. However, this means dealing with an issue we are trying to avoid as much as possible: setting hard thresholds in color space. As an alternative way to improve the false positive rate, we propose a novel potential function. Rather than specifying a constant value for β , it will now depend on the relative occurrence of the particular clique in a foreground (fire) and background model. Note that this means a departure from the MLL theory. For every possible pair of color labels, a potential value is now calculated beforehand, based on training data. Let $C_f(f_i, f_j)$ denote the number of times a pairwise clique consisting of the labels f_i and f_j occurred over all fire areas in the training data, and likewise $C_b(f_i, f_j)$ the number of times it occurred over all background areas. We will then estimate the occurrence probabilities of the clique in foreground and background as

$$P_f(f_i, f_j) = \frac{1 + C_f(f_i, f_j)}{\sum \sum_{f_i, f_j \in L} C_f(f_i, f_j) + |L|^2} \quad (8)$$

$$P_b(f_i, f_j) = \frac{1 + C_b(f_i, f_j)}{\sum \sum_{f_i, f_j \in L} C_b(f_i, f_j) + |L|^2} . \quad (9)$$

Note that we added 1 to the occurrence counts of each clique to avoid probabilities of zero, as is common practice (e.g. for training a Bayes classifier). This gives rise to the term $|L|^2$ in the denominator. The potential function V_2 we propose is then given by

$$V_2(f_i, f_{i'}) = \begin{cases} \frac{P_b(f_i, f_j)}{P_b(f_i, f_j) + P_f(f_i, f_j)} & |f_i = f_j \\ -\frac{P_f(f_i, f_j)}{P_b(f_i, f_j) + P_f(f_i, f_j)} & |f_i \neq f_j \end{cases} \quad (10)$$

The value in the first case is the probability that, if this particular clique occurs, it is caused by the background model. Likewise the value in the second case is the probability that it is caused by the fire model. While this potential function is obviously heuristic, it implements the functionality we require:

- cliques of uniform color are penalized, but more so for unlikely fire colors,
- cliques of different color are encouraged, but more so for typical fire combinations.

Experiments show that with the new potential function V_2 , the areas near the edges of flames generate very low energy, while the entire background results in much higher energy values. The interior part of the flame falls in between, on average generating more energy than the flame edge but less than the background.



Fig. 2. Source frame, detected blocks with high energy threshold, and corresponding component with lower energy threshold

This is exploited in a two-stage classifier. The energy function is calculated on 4×4 pixel blocks and first thresholded on an energy level T_1 allowing the entire fire areas to pass the criterion, as well as some spurious detections in the non-fire areas. In the second stage, the calculated energy is thresholded on a level $T_2 < T_1$, allowing only series of blocks near the edges of the flames. Only the first-stage connected components which contain blocks from the second stage are retained, resulting in much fewer false detections. An example can be seen in figure 2.

4 Performance

The color occurrence probability distributions P_f and P_b were trained on a set of 6 ground truth images of fire, and 6 additional background images featuring a variety of settings. The fire images are video frames depicting four different fires, captured by different types of cameras and from different viewing angles. The images also exhibit a wide variety of camera settings, from underexposure to oversaturation and varying degrees of focal sharpness. The additional background images were included for training balance, as the video frames were all captured in an industrial environment and therefore featured similar backgrounds. This training is intended to be universal, so no retraining is required for use in different circumstances. However, results may improve further for very specific scenarios when the method is trained on the according scenario-specific imagery.

The performance of the fire detection system was evaluated on over 49,000 video frames and compared to the fire detection method proposed by Celik *et al.* [14], which defines a set of rules in Cr-Cb color space based on three

Table 1. Performance statistics of Celik *et al.* compared to the proposed method

Method	Detection rate	False alarm rate
Celik 2008	99.95%	50.93%
Proposed (first stage only)	99.98%	42.82%
Proposed (first and second stage)	99.57%	21.80%

polynomial curves. The method was implemented as described in the paper, taking care to use the same 8-bit range for the chroma planes. The results were also aggregated into 4x4 blocks using a majority voting rule, to make comparison with our method as fair as possible. We consider this method to be the state-of-the-art single-pixel fire color filter against which to judge the benefits of our contextual modeling.

The first part of the test set consists of 30,000 frames depicting fire, to obtain the detection rate. These video frames show a number of controlled fires in an outdoor firemen training complex built to resemble an industrial site. The fires include a burning petroleum tank, a ruptured gas pipe, a round tank engulfed in flames and a fire in a maintenance trench. The fires were monitored by six cameras of different types, placed on different elevation levels and angles. The fire is considered detected as soon as at least one of its pixel blocks is detected as a fire block. In the interest of fairness, we should note that the training images for our method were captured on the same site, albeit at a different time with different sunlight levels.

The second part of the test set contains over 18,000 video frames captured from a moving vehicle in an urban environment. This set is representative of the occurrence of fire-colored objects to be expected in the busiest environments, e.g. red and yellow clothing, vehicles or advertising. An image is counted as a false positive when one or more blocks in the image are classified as fire.

The results obtained on this data set are shown in Table II. The reference method by Celik *et al.* scored a detection rate of 99.95% on the fire frames, while generating over 50% false positives. This illustrates the high occurrence of fire-colored objects in the second dataset: over half of the frames contain at least one fire-colored 4x4 block. In comparison, the detection rate of our proposed method after just the first stage was 99.98%, with a false detection rate of 42.82%. This shows that even after just the least discriminative of the two stages, there is an improvement over the reference method. After both stages of the method, the detection rate drops only slightly to 99.57%, while false positives are much reduced to 21.80%. These statistics prove the adequacy of the system as standalone fire detector. The cases in which the fire was not detected are mostly transition phases, either just after the fire was started or when it was nearly extinguished. One can reasonably assume that any spreading fire will be detected. The false negatives can thus be considered rare and temporary manifestations of fire in which the spectral texture is coincidentally and atypically low.

5 Conclusion

We have designed an MRF-based visual fire detection system which is easy to train, and requires optimization of just one critical parameter (the lower classifier energy threshold) rather than setting multiple fixed color rules. Furthermore, after training on basic, generic ground-truth data the method is proven to yield very good detection rates in a variety of circumstances, while at the same time significantly reducing false positives over standard color-based methods. Moreover, it does not rely on any temporal information, and can therefore be applied

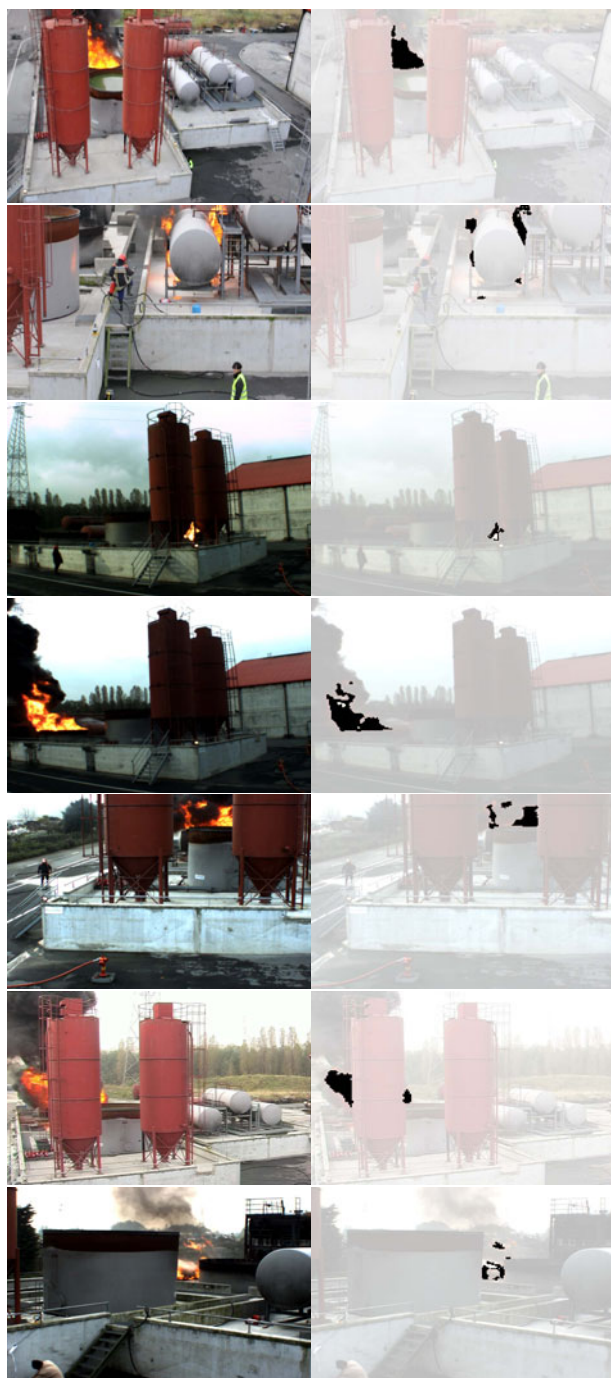


Fig. 3. Examples frames and their detector output

to still images and low framerate cameras without performance degradation. On the other hand, if normal video frame rates and sufficient computing power are available, the method could be improved further by implementing temporal hysteresis, whereby multiple subsequent alerts are required before the alarm is set off. The model uses insignificant amounts of memory (typically 256 bytes) and the block-based processing suits parallel implementation, making the method ideal for implementation on dedicated hardware (e.g. FPGAs) to speed up computation.

References

1. Bao, H., Li, Y., Zeng, X.Y., Zhang, J.: A fire detection system based on intelligent data fusion technology. In: International Conference on Machine Learning and Cybernetics, vol. 2, pp. 1096–1101 (2003)
2. Li, Y., Vodacek, A., Kremens, R.L., Ononye, A., Tang, C.: A hybrid contextual approach to wildland fire detection using multispectral imagery. *IEEE Transactions on Geoscience and Remote Sensing* 43, 2115–2126 (2005)
3. Healey, G., Slater, D., Lin, T., Drda, B., Goedeke, A.: A system for real-time fire detection. In: CVPR, pp. 605–606 (1993)
4. Liu, C.-B., Ahuja, N.: Vision based fire detection. In: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR 2004), Washington, DC, USA, vol. 4, pp. 134–137. IEEE Computer Society Press, Los Alamitos (2004)
5. Töreyn, B.U., Dedeoglu, Y., Güdükbay, U., Çetin, A.E.: Computer vision based method for real-time fire and flame detection. *Pattern Recogn. Lett.* 27, 49–58 (2006)
6. Celik, T., Demirel, H., Ozkaramanli, H., Uyguroglu, M.: Fire detection using statistical color model in video sequences. *J. Vis. Commun. Image Represent.* 18, 176–185 (2007)
7. Huang, P.H., Su, J.Y., Lu, Z.M., Pan, J.S.: A fire-alarming method based on video processing. In: International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 359–364 (2006)
8. Ko, B., Hwang, H.J., Lee, I.G., Nam, J.Y.: Fire surveillance system using an omnidirectional camera for remote monitoring. In: Proceedings of the 2008 IEEE 8th International Conference on Computer and Information Technology Workshops, CITWORKSHOPS 2008, Washington, DC, USA, pp. 427–432. IEEE Computer Society Press, Los Alamitos (2008)
9. Töreyn, B.U., Çetin, A.E.: Online detection of fire in video. In: CVPR (2007)
10. Zhang, D., Shizhong, H., Zhao, J., Zhang, Z., Chengzhang, Q., Youwang, K., Xiang, C.: Image based forest fire detection using dynamic characteristics with artificial neural networks. In: International Joint Conference on Artificial Intelligence, pp. 290–293 (2009)
11. Kandil, M., Salama, M.: A new hybrid algorithm for fire vision recognition. In: EUROCON 2009, pp. 1460–1466. IEEE, Los Alamitos (2009)
12. Zhang, Z., Zhao, J., Zhang, D., Qu, C., Ke, Y., Cai, B.: Contour based forest fire detection using fft and wavelet. In: Proceedings of the 2008 International Conference on Computer Science and Software Engineering, CSSE 2008, Washington, DC, USA, pp. 760–763. IEEE Computer Society Press, Los Alamitos (2008)

13. Noda, S., Ueda, K.: Fire detection in tunnels using an image processing method. In: Proceedings Vehicle Navigation and Information Systems Conference, pp. 57–62 (1994)
14. Celik, T., Kai-Kuang, M.: Computer vision based fire detection in color images. In: IEEE Conference on Soft Computing in Industrial Applications, pp. 258–263 (2008)
15. Xiang, Y., Zhou, X., Chua, T., Ngo, C.: A revisit of generative model for automatic image annotation using markov random fields. In: CVPR, pp. 1153–1160 (2009)
16. Chan, A.B., Vasconcelos, N.: Layered dynamic textures. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31, 1862–1879 (2009)
17. Li, S.Z.: *Markov Random Field Modeling in Computer Vision*. Springer, Heidelberg (1995)

A Virtual Curtain for the Detection of Humans and Access Control

Olivier Barnich, Sébastien Piérard, and Marc Van Droogenbroeck

INTELSIG Laboratory, Montefiore Institute, University of Liège, Belgium

Abstract. Biometrics has become a popular field for the development of techniques that aim at recognizing humans based upon one or more intrinsic physical or behavioral traits. In particular, many solutions dedicated to access control integrate biometric features like fingerprinting or face recognition.

This paper describes a new method designed to interpret what happens when crossing an invisible vertical plane, called *virtual curtain* hereafter, at the footstep of a door frame. It relies on the use of two laser scanners located in the upper corners of the frame, and on the classification of the time series of the information provided by the scanners after registration. The technique is trained and tested on a set of sequences representative for multiple scenarios of normal crossings by a single person and for tentatives to fool the system.

We present the details of the technique and discuss classification results. It appears that the technique is capable to recognize many scenarios which may lead to the development of new commercial applications.

1 Introduction

Detecting a person, locating him, and recognizing its identity are three cornerstones of applications turned on security. Over the past years, many technologies based on biometrical signatures have emerged to achieve these goals. The handbook by Jain *et al.* [1] illustrates the many techniques available today. They ranges from fingerprinting, voice recognition, face detection, dental identification techniques to iris, gesture or gait recognition, just to name a few.

In this paper, we propose a new platform (comprising hardware and software) for critical applications such as secure access control, where biometrics has become a viable technology, that can be integrated in identity management systems. Commonly, access to restricted areas is monitored by a door with an electrical lock or a revolving door activated by the swipe of an access control card. In this context, we aim for a system able to send an alarm when the expected scenario of a single person crossing the door frame is not confirmed; this could occur when someone enters a restricted area by passing through the door at the same time as another person (this is called *piggybacking* when the other person is authorized and *tailgating* when the other person is unauthorized).

The purpose of our method is to identify the scenario of one or more persons when they cross a door frame. While camera driven solutions exist for it, we

deliberately chose to rely on laser scanners instead because they can be directly embedded in a door frame and do not require a controlled environment. In addition, the complete solution is required to operate in real time with a limited amount of processing power, and it appeared that the choice of laser scanners proved adequate, retrospectively.

The article by Sequeira *et al.* [2] is representative for the problems faced with laser scanners. Some of them are:

- occlusions and shadows. Objects located beyond other objects are not detected (occlusion) and, likewise, it is impossible to interpret the scene beyond the first object reached by the rays of a laser (shadowing).
- angle of acquisition. Laser scanners have a narrow aperture angle in one or two dimensions. Also, as the technology is based on a sender/receiver mechanism, physical properties and the angle of incidence are important.
- scan overlap. To interpret fast movements, data has to be captured either at a high speed, or with a fair overlap between successive scans. A practical solution consists in the use of linear scanners with a high acquisition rate.
- scan resolution. Radial scanners output distances of closest objects for a fixed set of angles. The radial sampling might be uniform, this does not mean that the precision on the distance both in the direction of the light ray or in the direction perpendicular to it is uniform as well. In fact, objects should be closer to increase the measurement accuracy, but the price to pay is an increased shadowing effect.

The paper is organized as follows. As we propose an original placement of laser scanners and a new method to build a virtual curtain (which is an invisible and immaterial membrane), we first describe our set-up in Section 2. From this arrangement of scanners, we derive the notion of a virtual curtain, described in Section 2.2. This concept is the key of a classification process detailed in Section 3. First, surfacic features are extracted from the intersection of the curtain and an object or a person that crosses it. Then we concatenate these features over time to derive a windowed temporal signature. This signature is then used to identify the scene by a classification process; the purpose is to raise an alarm when the normal situation of a single person crossing the curtain is not met, for example when several persons want to pass the door simultaneously. Results of this classification method obtained over a database of more than 800 sequences are provided in Section 4. Section 5 concludes the paper.

2 Original Set-Up

A real security application requires that the system is insensitive to lighting conditions. Consequently, we cannot afford using a background subtracted video stream to recover the binary silhouettes of the walkers. Instead, we use laser devices described in the next section. Then we develop the concept of virtual curtain.

2.1 Sensors

Laser range sensors are widely used nowadays. Simple devices measure distances for a few 3D directions, but more sophisticated sensors exist. For example, there are devices that are used in conjunction with rotating mirrors to scan a 360° field of view. For our platform, we use the rotating laser sensors manufactured by B.E.A. (see Figure 1).



Fig. 1. The laser sensor used in our experiments (the LZR P-200 manufactured by B.E.A. S.A., <http://www.bea.be>)

These laser range sensors are completely independent of the lighting conditions, as they rely on their own light sources. They are able to measure the distance between the scanner and surrounding objects by sending and receiving laser pulses in a plane. The measurement process is discrete; it samples the angles with an angular precision of 0.35° and covers an angular aperture of 96°. The plane is scanned 60 times per second. In practical terms, these sensors deliver a signal $d_t(\theta)$ where (1) d is the distance between the sensor and the object hit by the laser ray, (2) θ denotes the angle in the scanning plane ($0 \leq \theta \leq 96^\circ$), and (3) t ($= \frac{k}{60}s$ for $k = 0, 1, 2, \dots$) is the time index.

The information that these sensors provide has a physical meaning (the distance is given in millimeters) and relates only to the geometrical configuration; the color and texture of objects have no impact on the measurements. Note also that these sensors have been designed to be integrated in difficult industrial environments, like revolving doors, where a camera might not fit as well.

2.2 Towards the Concept of Virtual Curtain

Theoretically, a single scanner suffices to build a 2D shape. However, we have decided to use two scanners to reduce the shadowing effects resulting from a single scanner. The sensing system is made of two laser scanners located in the two upper corners of the frame of a door (see Figure 2). Consequently, distances are measured in a plane that comprises the vertical of the gravity and the straight line joining the two sensors. This is our concept of *virtual curtain*. To some extent, it can be seen as a wide traversable waterfall, except that you don't get wet if you cross it!

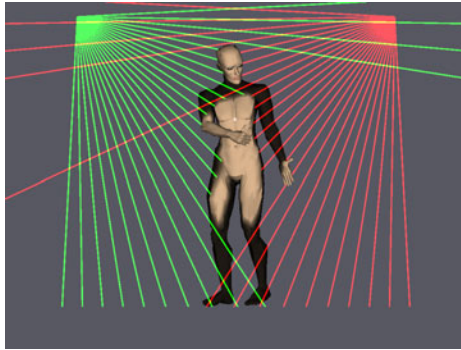


Fig. 2. Arrangement of the sensors. The two rotating laser range sensors are located in the upper corners of a vertical plane. They send rays that hit the lateral door frame, the ground or a person crossing the plane.

It must be noted that signals collected from the two sensors are not synchronized. This impacts on the system. A detailed analysis of the physical uncertainty resulting from desynchronization shows that, in the worst case, a shift of 10 cm at the height of the knee is possible. As a matter of fact, the physical precision on the location of a point increases with its height in the reconstructed plane. In other words, the horizontal imprecision, due to desynchronization, decreases with the height of a point.

Furthermore, as measures correspond to the distance between the sensor and the first point hit by the laser along its course, they account for a linear information related to the central projection of the silhouette of the moving objects passing through the door; it implies that points located beyond the first point are invisible and that widths are impossible to measure with a single scanner. With two scanners, there are less ambiguities but some of them remain, for example in the bottom part of the silhouette. Furthermore, a hole in the silhouette cannot be detected. In practice, the subsequent classification algorithm has to be robust enough to be able to deal with these ambiguities.

2.3 Computing a Virtual Curtain

We now describe how to build a series of silhouettes of a walking human crossing the virtual curtain.

Polar transformation and registration of the two signals. Since the information given by the laser scanners is polar, the first step towards the reconstruction of an image related to the shape of the scanned objects is a polar transformation of the raw signals:

$$x_t(\theta) = d_t(\theta) \cos(\theta), \quad (1)$$

$$y_t(\theta) = d_t(\theta) \sin(\theta). \quad (2)$$

Every $\frac{1}{60}s$, we get two signals $d_t(\theta)$ covering the quarter area of a plane, one per sensor, and apply a polar transformation to them. Then we connect successive points with straight line segments and register the two signals according to the width of the frame of the door. This process is illustrated in Figure 3.

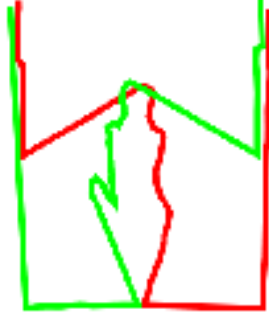


Fig. 3. Construction of a curve per sensor that corresponds to the closest visible points. The signal captured by the left (right) sensor is displayed in green (red).

Thanks to the calibration of the sensors and the real physical distances they deliver, the registration process is simple as it relies exclusively on the physical dimensions of the door. Note however that since the signals provided by the two sensors are not synchronized, the registration of the sensor signals will be affected by a time jitter that impacts on the overall signal to noise ratio.

Flood fill and intersection. For each laser scanner, we now have a continuous line that outlines one side of the silhouette of the object seen in the curtain. We still need to reconstruct one complete silhouette. The reconstruction of a half silhouette is achieved by closing the contour and applying a flood fill algorithm to the continuous line that outlines it. Then the two half silhouettes are intersected to get the silhouette. The reconstruction process is illustrated in Figure 4.

We can see that the upper parts of the silhouette (in principle, the shoulders and the head) are better represented than the lower parts of the silhouette because the lasers are closer to the upper part and thus do sample this part of the shape with a higher precision. As a matter of fact, the legs are almost absent from the reconstructed silhouettes. Furthermore, we showed in Section 2.2 that the lack of synchronization of the sensors causes an horizontal imprecision that decreases with the height of a point.

The reconstruction of a silhouette happens 60 times per second. Figure 5 shows a 3D volume obtained by piling up the successive silhouettes of one (left-hand side picture) or two (right-hand side picture) persons while they cross the curtain. This 3D shape is not very intuitive. Therefore we need to elaborate on the appropriate features to describe it.

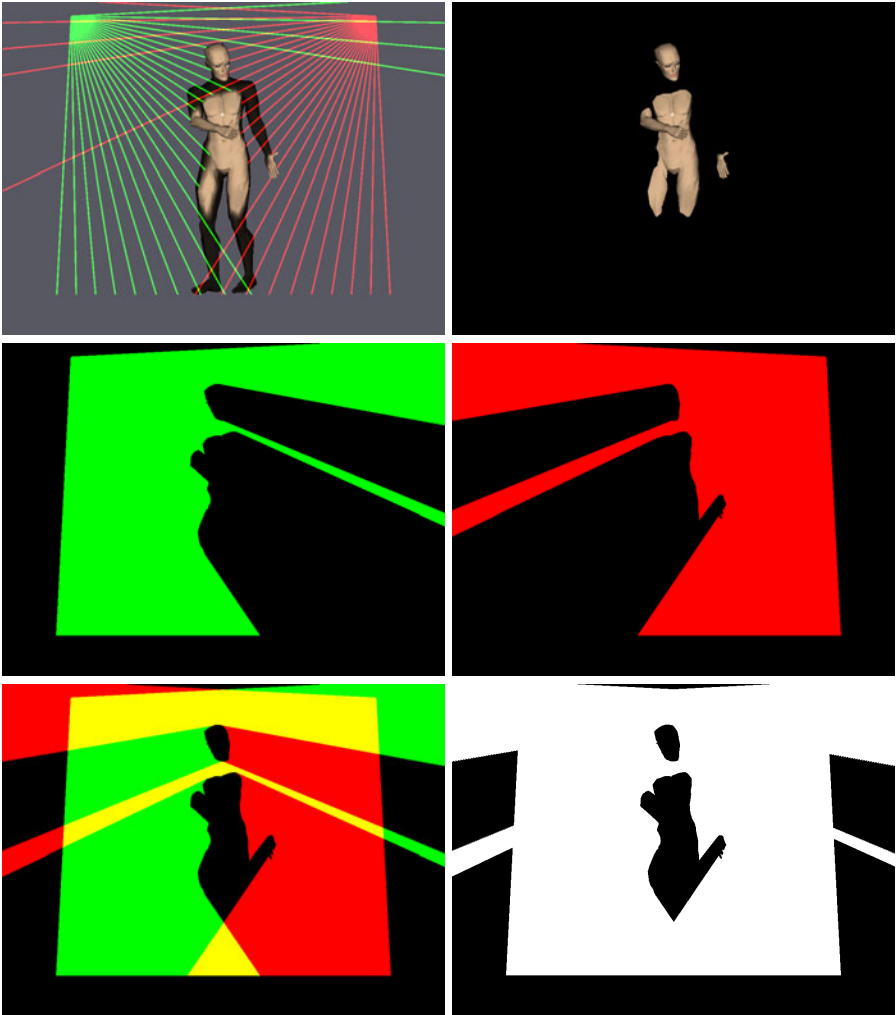


Fig. 4. Illustration of the silhouette reconstruction process

3 Features Extraction and Classification

The closest field to our application in the literature is that of human gait recognition where relevant features are extracted from the time series of the binary silhouettes of a moving object. Classification is then performed on the basis of these features (often referred to as *signature*) to recover the identity of the walking human in front of the camera. In our application, we can use these features extraction and classification techniques to identify the time series of silhouettes that correspond to a *single* walking human from the others.

A good introductory reading about gait recognition can be found in [3,4,5]. An extensive review of the existing techniques is presented in [6].

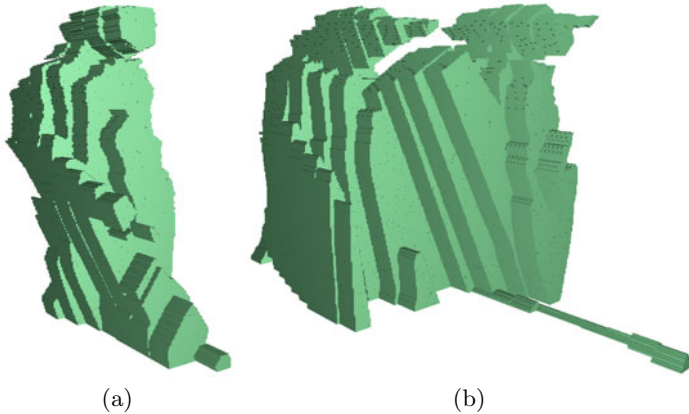


Fig. 5. Volume whose slices are consecutive silhouettes of one person (a) or two persons (b) crossing the curtain over time

Model-based approaches process sequences of images to estimate the parameters of an explicit gait model. These estimated values are then used to recover the identity of the walking human. It should be noted that, in our case, these silhouettes need to be reconstructed from the contour-related information given by the two radial sensors.

These methods often need high definition images in order to work properly which is a major drawback for our application since the laser sensors only provide 274 points per scan for an angular aperture of 96° . Furthermore, they exhibit a significantly higher computational cost than silhouette-based techniques. This is also a serious issue since real-time processing is required in our application.

Silhouette-based approaches do not rely on any explicit model for the walking human(s). These techniques extract signatures directly from series of silhouettes. A simple approach is described in [7] where the areas of raw (re-sized) silhouettes are used as a gait signature. In [8], the gait template of a walking human is computed by averaging the corresponding binary silhouettes. The classification is then achieved using a nearest neighbor technique.

The contours of silhouettes have been used in [9] and by Soriano *et al.* in [10] where signatures are derived from series of Freeman encoding of the re-sized silhouette shape. An angular transform of the silhouette is proposed in [11] and is said to be more robust than the raw contour descriptions.

The gait signature of [12] is based on horizontal and vertical projections of the silhouettes. The authors of [13] consider time series of horizontal and vertical projections of silhouettes as *frieze* patterns. Using the framework of frieze patterns, they estimate the viewing direction of the walking humans and align walking sequences from similar viewpoints both spatially and over time. Cross-correlation and nearest neighbor classification is then used to perform the identification of the walkers. To get an increased robustness to differences between the training and test sets, [14] proposes a technique that relies on frieze patterns of frame differences between a key silhouette and a series of successive silhouettes.

In our application, we apply an approach similar to silhouette-based gait recognition. First we extract surfacic features from a single silhouette. Then we aggregate features over time to obtain a temporal signature that is used to identify the ongoing crossing scenario.

3.1 Feature Extracted from the Intersection between an Object and the Virtual Curtain

To characterize reconstructed silhouettes, we use the notion of *cover by rectangles*. The cover by rectangles is a morphological descriptor defined as the union of all the largest rectangles that can fit inside of a silhouette. The whole idea is described in [15].

From the cover of a silhouette, many features can be extracted to build a silhouette signature. Features that could be considered to characterize the dataset are:

- The set of the enclosed rectangles (that is, the cover itself).
- The maximum width (or height) of the rectangles included in the cover.
- Histogram of the widths (or heights) of the rectangles included in the cover.
- 2D histogram of the widths and heights of the rectangles included in the cover.
- The horizontal or vertical profile of the silhouette.

Due to the unusual shape of the silhouette, there is no prior art about the best suited characteristics. Therefore, we fall back to proved intra-frame signatures that were considered in [16] for gait recognition. They are:

- The 2D histogram of the widths and heights of the rectangles included in the cover (denoted as $\mathcal{G}^{W \times H}(i, j)$), and
- The concatenation of the histogram of the widths and the histogram of the heights of the rectangles included in the cover (denoted as $\mathcal{G}^{W+H}(i, j)$).

Note that in order to build histograms, we partition the widths and heights of the rectangles respectively into M bins and N bins. The best values for M , N are discussed later.

Temporal features. The full signature is constructed as a combination of intra-frame silhouette signatures. Its purpose is to capture the time dynamics of the moving object crossing the door. In our application, the time dynamics may be very important. One of the proposed solution to handle the temporal evolution of a shape is to normalize the gait cycle, like in [17].

Like for gait recognition, this poses a problem in that the classifier will delay its answer until the end of the sequence. An alternative solution is to normalize the sequence by parts. Another approach to consider is to learn several speeds during the database set-up, and use the global normalization as a fallback or confirmation step.

Our approach is much simpler and provides results similar to results obtained with other approaches. Our inter-frame spatio-temporal signature (denoted as

$\mathcal{G}^{W+H}(i, j, t)$ or $\mathcal{G}^{W \times H}(i, j, t)$) is the *concatenation* of a given number L of consecutive intra-frame signatures.

3.2 Classification

Classification consists in the learning of a function from labeled input data. The learned function, sometimes referred to as *model*, is later used to predict the label of unlabeled data. The purpose of the classifier is to *generalize* the knowledge present in the labeled examples to new samples.

When a person crosses the virtual curtain, we reconstruct the time series of his binary silhouettes and assign a class to it with our classification algorithm. In this particular application, only two classes can be assigned to a series of silhouettes:

- “0”, which denotes that a single person has crossed the virtual curtain,
- and “1”, which denotes that more than one person have crossed the virtual curtain.

Learning and cross-validation. To build a classifier, it is necessary to label (manually) a large amount of data samples. Part of these labeled samples are used to train the classifier. They constitute the “*learning set*”. Remaining labeled samples are used to evaluate the performances of the classifier; they are part of the “*test set*”.

A rule of thumb is to divide the available labeled data in two equal parts: one to train the model, and the other to test it. With only a few available labeled data, it may be disadvantageous to ignore half of the labeled data to train the model. A common solution is then to use cross-validation briefly described hereafter.

If there are N labeled samples, cross-validation consists in dividing them into K subsets of equal size. $(K - 1)$ subsets are used to train the model while the remaining one is used to test it. This procedure is repeated K times, for each test set on turn. The final score of the classifier is the average of the K computed scores. When $K = N$, this method is called *leave-one-out*.

Classification tool. There are many classification techniques available. Among the most popular are nearest neighbors classifiers (KNN), artificial neural networks (ANN), (ensemble of) decision trees, and support vector machines (SVM).

In our case, the sets of features extracted from the time series of reconstructed silhouettes are classified with a support vector machine classifier [18]. An SVM is a binary classifier that maps its input features into a high-dimensional non-linear subspace where a linear decision surface is constructed. We used the implementation provided by `libsvm` [19] with a radial basis function (RBF) kernel.

4 Results

To evaluate the performances of our algorithm, the B.E.A. company has provided us 349 labeled sequences of a single person (class “0”) and 517 sequences

that contain two walkers (class “1”). We use these sequences to build databases of labeled signatures for different sets of parameters (M , N , and L are the parameters, that is the number of bins for the rectangle widths, the number of bins for the rectangle heights, and the number of intra-frame features aggregated in a signature respectively). For each set of parameters, we employ 5-fold cross-validation on the corresponding database to assess the precision of the classification according to the error rate (E) defined by

$$E = \frac{FP + FN}{TP + TN + FP + FN}, \quad (3)$$

where TP is the number of true positives, TN the number of true negatives, FP the number of false positives, and FN the number of false negatives.

We tested both $\mathcal{G}^{W+H}(i, j, t)$ and $\mathcal{G}^{W \times H}(i, j, t)$. The two corresponding series of results are given in Table 1 and Table 2.

Table 1. Error rates obtained for $\mathcal{G}^{W+H}(i, j, t)$

E [%]	$M = N = 2$	$M = N = 4$	$M = N = 6$	$M = N = 8$	$M = N = 10$
$L = 40$	15.99	14.43	14.37	14.68	14.76
$L = 60$	9.77	8.34	8.23	8.86	8.55
$L = 70$	7.89	7.04	6.70	6.86	7.17
$L = 80$	6.98	6.05	5.68	6.19	5.91
$L = 90$	7.51	7.65	7.44	7.09	7.51
$L = 100$	9.84	11.58	10.04	10.86	10.97
$L = 120$	18.15	18.15	18.16	16.34	17.50

Table 2. Error rates obtained for $\mathcal{G}^{W \times H}(i, j, t)$

E [%]	$M = N = 2$	$M = N = 4$	$M = N = 6$	$M = N = 8$	$M = N = 10$
$L = 40$	16.06	14.05	14.07	14.04	14.76
$L = 60$	9.83	8.96	9.79	10.25	10.79
$L = 70$	8.01	7.82	8.35	8.35	8.57
$L = 80$	7.12	7.45	7.5	7.26	7.64
$L = 90$	8.14	8.62	8.62	8.76	8.62
$L = 100$	10.45	9.43	9.84	9.73	11.79
$L = 120$	19.81	17.33	16.01	16.01	18.32

They show that an error rate as low as 5.68% is reached for the $\mathcal{G}^{W+H}(i, j, t)$ signature with $M = N = 6$, and $L = 80$. We also observe that, for this particular problem, L is the parameter with the largest variability in the result. From our tests, the best results are obtained for a signature length L of 80 frames, a number that matches the average time to cross the curtain. For M and N , the choice of a value is less critical but, from our tests, it appears that $M = N = 4$ is an appropriate choice. We also noticed that for this particular problem, $\mathcal{G}^{W+H}(i, j, t)$ has slightly better results than $\mathcal{G}^{W \times H}(i, j, t)$ while having a reduced computational cost. One explanation to this is that the shadowing effect in the lower part of the silhouette adds more noise on $\mathcal{G}^{W \times H}(i, j, t)$ than on $\mathcal{G}^{W+H}(i, j, t)$.

Finally, it must be noted that during our tests, we observed that for low global error rates, the number of FN is considerably lower than the number of FP. In other words, the system is naturally more inclined to reject a single person than to allow to a group of two persons to pass the door. For an access control system, it is a welcomed property.

5 Conclusions

This paper introduces the concept of virtual curtain that is obtained by the registration of two linear laser scanners that measure distances in a same plane. Despite intrinsic shortcomings, originated by effects like occlusion or shadowing, features derived from an object crossing a virtual curtain permit to interpret the scene. In particular, it is shown how it is possible to differentiate between several scenarios for the context of access control. Features are first extracted for every intersection between an object and the curtain, then they are concatenated to provide a temporal signature. This signature is handled by a classification process that identifies the ongoing scenario. Results show that a high recognition rate is achievable for a pre-defined set of training and testing scenarios. In practice, we will have to wait for international standardization bodies or organizations to elaborate some criteria to benchmark the performances for a use under variable operational conditions. But our results proof that our system is tractable and usable for the interpretation of a scene.

Acknowledgments. This work was supported by the Belgian Walloon Region (<http://www.wallonie.be>). S. Piérard has a grant funded by the FRIA (<http://www.frs-fnrs.be/>). We are also grateful to *B.E.A. S.A.* (<http://www.bea.be>), and in particular to Y. Borlez, O. Gillieux, and E. Koch, and for their invaluable help.

References

1. Jain, A., Flynn, P., Ross, A.: Handbook of Biometrics. Springer, Heidelberg (2008)
2. Sequeira, V., Boström, G.: Gonçalves, J.: 3D site modelling and verification: usage of 3D laser techniques for verification of plant design for nuclear security applications. In: Koschan, A., Pollefeys, M., Abidi, M. (eds.) 3D Imaging for Safety and Security, pp. 225–247. Springer, Heidelberg (2007)
3. Boulgouris, N., Hatzinakos, D., Plataniotis, K.: Gait recognition: a challenging signal processing technology for biometric identification. *IEEE Signal Processing Magazine* 22(6), 78–90 (2005)
4. Nixon, M., Carter, J., Shutler, J., Grant, M.: New advances in automatic gait recognition. *Elsevier Information Security Technical Report* 7(4), 23–35 (2002)
5. Nixon, M.: Gait biometrics. *Biometric Technology Today* 16(7-8), 8–9 (2008)
6. Nixon, M., Tan, T., Chellappa, R.: Human identification based on gait. Springer, Heidelberg (2006)
7. Foster, J., Nixon, M., Prügel-Bennett, A.: Automatic gait recognition using area-based metrics. *Pattern Recognition Letters* 24(14), 2489–2497 (2003)

8. Huang, X., Boulgouris, N.: Human gait recognition based on multiview gait sequences. *EURASIP Journal on Advances in Signal Processing*, 8 (January 2008)
9. Wang, L., Tan, T., Ning, H., Hu, W.: Silhouette analysis-based gait recognition for human identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25(12), 1505–1518 (2003)
10. Soriano, M., Araullo, A., Saloma, C.: Curve spreads: a biometric from front-view gait video. *Pattern Recognition Letters* 25(14), 1595–1602 (2004)
11. Boulgouris, N., Chi, Z.: Gait recognition using radon transform and linear discriminant analysis. *IEEE Transactions on Image Processing* 16(3), 731–740 (2007)
12. Kale, A., Cuntoor, N., Yegnanarayana, B., Rajagopalan, A., Chellappa, R.: Gait analysis for human identification. In: *Proceedings of the International Conference on Audio-and Video-Based Person Authentication*, Guildford, UK, pp. 706–714 (2003)
13. Liu, Y., Collins, R., Tsin, Y.: Gait sequence analysis using frieze patterns. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) *ECCV 2002*. LNCS, vol. 2351, pp. 657–671. Springer, Heidelberg (2002)
14. Lee, S., Liu, Y., Collins, R.: Shape variation-based frieze pattern for robust gait recognition. In: *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8 (June 2007)
15. Barnich, O., Jodogne, S., Van Droogenbroeck, M.: Robust analysis of silhouettes by morphological size distributions. In: Blanc-Talon, J., Philips, W., Popescu, D., Scheunders, P. (eds.) *ACIVS 2006*. LNCS, vol. 4179, pp. 734–745. Springer, Heidelberg (2006)
16. Barnich, O., Van Droogenbroeck, M.: Frontal-view gait recognition by intra- and inter-frame rectangle size distribution. *Pattern Recognition Letters* 30(10), 893–901 (2009)
17. Boulgouris, N., Plataniotis, K., Hatzinakos, D.: Gait recognition using linear time normalization. *Pattern Recognition* 39(5), 969–979 (2006)
18. Cortes, C., Vapnik, V.: Support-vector networks. *Machine Learning* 20(3), 273–297 (1995)
19. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001) Software available at, <http://www.csie.ntu.edu.tw/~cjlin/libsvm>

A New System for Event Detection from Video Surveillance Sequences

Ali Wali, Najib Ben Aoun, Hichem Karray, Chokri Ben Amar,
and Adel M. Alimi

REGIM: REsearch Group on Intelligent Machines, University of Sfax, National
School of Engineers (ENIS), BP 1173, Sfax, 3038, Tunisia

{ali.wali,najib.benaoun,hichem.karray,chokri.benamar,adel.alimi}@ieee.org

Abstract. In this paper, we present an overview of a hybrid approach for event detection from video surveillance sequences that has been developed within the REGIMVid project. This system can be used to index and search the video sequence by the visual content. The platform provides moving object segmentation and tracking, High-level feature extraction and video event detection. We describe the architecture of the system as well as providing an overview of the descriptors supported to date. We then demonstrate the usefulness of the toolbox in the context of feature extraction, events learning and detection in large collection of video surveillance dataset.

1 Introduction

Image and video indexing and retrieval continue to be an extremely active area within the broader multimedia research community [3,17]. Interest is motivated by the very real requirement for efficient techniques for indexing large archives of audiovisual content in ways that facilitate subsequent usercentric accessing. Such a requirement is a by-product of the decreasing cost of storage and the now ubiquitous nature of capture devices. The result of which is that content repositories, either in the commercial domain (e.g. broadcasters or content providers repositories) or the personal archives are growing in number and size at virtually exponential rates. It is generally acknowledged that providing truly efficient usercentric access to large content archives requires indexing of the content in terms of the real world semantics of what it represents.

Furthermore, it is acknowledged that real progress in addressing this challenging task requires key advances in many complementary research areas such as; scalable coding of both audiovisual content and its metadata, database technology and user interface design. The REGIMVid project integrates many of these issues (fig.1). A key effort within the project is to link audio-visual analysis with concept reasoning in order to extract semantic information. In this context, high-level pre-processing is necessary in order to extract descriptors that can be subsequently linked to the concept and used in the reasoning process. In addition to concept-based reasoning, the project has other research activities that require high-level feature extraction (e.g. semantic summary of metadata

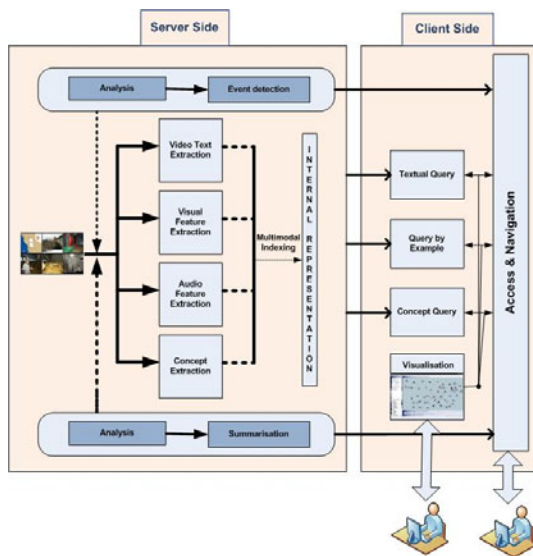


Fig. 1. REGIMVid platform Architecture

[5], Text-based video retrieval [10,6], event detection [16] and Semantic Access to Multimedia Data [12]) it was decided to develop a common platform for descriptor extraction that could be used throughout the project. In this paper, we describe our subsystem for video surveillance indexing and retrieval. The remainder of the paper is organised as follows: a general overview of the toolbox is provided in Section 2, include a description of the architecture. In section 3 we present our approach to detect and extract of moving objects from video surveillance dataset. It includes a presentation of different concepts taken care by our system. We present the combining single SVM classifier for learning video events in section 4. The descriptors of the visual feature extraction will be presented in section 5. Finally, we present our experimental results for both event and concept detection future plans for both the extension of the toolbox and its use in different scenarios.

2 Our System Overview

In this section, we present an overview of the structure of the toolbox. The system currently supports extraction of 10 low-level (see section 5) visual descriptors. The design is based on the architecture of the MPEG-7 eXperimentation Model (XM), the official reference software of the ISO/IEC MPEG-7 standard.

The main objectif of our system is to provide automatic content analysis using concept/event-based and low-level features. The system (figure 2) first detect and segment the moving object from video surveillance dataset. In the second step, it extracts three class of features from the each frame, from a static background and the segmented objects(the first class from Ω_{in} , the second from Ω_{out} and

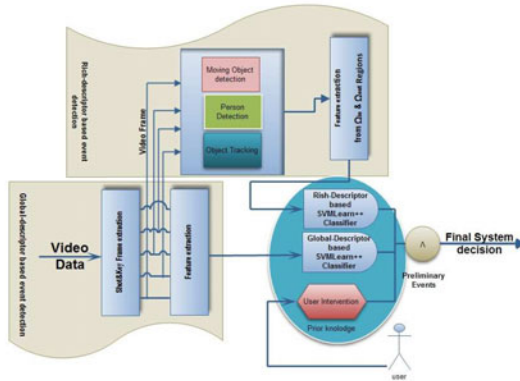


Fig. 2. Overview of our system for video input

the last class is from each key-frame in RGB color space, see subsection 3.2), and labels them based on corresponding features. For example, if three features are used (color, texture and shape), each frame has at least three labels from Ω_{out} , three labels from Ω_{in} and three labels from key-frame.

This reduces the video as a sequence of labels containing the common features between consecutive frames. The sequence of labels aim to preserve the semantic content, while reducing the video into a simple form. It is apparent that the amount of data needed to encode the labels is an order of magnitude lower than the amount needed to encode the video itself. This simple form allows the machine learning techniques such as Support Vector Machines to extract high-level features.

Our method offer a way to combine low-level features wish enhances the system performance. The high-level features extraction system according to our toolkit provides an open framework that allows easy integration of new features. In addition, the Toolbox can be integrated with traditional methods of video analysis. Our system offers many functionalities at different granularity that can be applied to applications with different requirements. The Toolbox also provides a flexible system for navigation and display using the low-level features or their combinations. Finally, the feature extraction according to the Toolbox can be performed in the compressed domain and preferably real-time system performance such as the videosurveillance systems.

3 Moving Object Detection and Extraction

To detect and extract a moving object from a video dataset we use a region-based active contours model where the designed objective function is composed of a region-based term and optimize the curve position with respect to motion and intensity properties. The main novelty of our approach is that we deal with the motion estimation by optical flow computation and the tracking problem simultaneously. Besides, the active contours model is implemented using a level

set, inspired from Chan and Vese approach [2], where topological changes are naturally handled.

3.1 Motion Estimation by Optical Flow

Recently, many motion estimation techniques were developed. Although, Block matching technique is the most used techniques and it have promising results motion estimation especially with improvement techniques [8], we have used the optical flow which had given us good results.

In our system, we use gradient-based optical flow algorithm proposed by Horn and Schunck [1]. similar to T. Macan and S. Loncaric [11], we have integrated the algorithm in multi-grid technique where the image is decomposed into Gaussian pyramid-set of the reduced images. The calculation starts at a coarser scale of the image decomposition, and the results are propagated to finer scales.

Let us suppose that the intensity of the image at a time t and position (x, y) is given by I (x, y, t). The assumption on brightness constancy is made that the total derivative of brightness function is zero which results the following equation:

$$\frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} + \frac{\partial I}{\partial t} = 0 \text{ or } I_{x,i}u_i + I_{y,i}v_i + I_{t,i} = 0 \tag{1}$$

This equation is named 'Brightness Change Constraint Equation'. Where u and v are components of optical flow in horizontal and vertical directions, respectively, and I_x, I_y and I_t are partial derivatives of I with respect to x, y and t respectively. Horn and Schunck added additional smoothness constraint because the equation (1) is insufficient to compute both components of optical flow. They minimized weighted sum of smoothness term and brightness constraint term:

$$\int_{\Omega} (I_x u + I_y v + I_t)^2 + \lambda (\|\nabla u\|^2 + \|\nabla v\|^2) dx \tag{2}$$

Minimization and discretization of equation (2) results in two equations for each image point where vector values u_i and v_i are optical flow variables to be determined. To solve this system of differential equations, we use the iterative Gauss-Seidel relaxation method.

3.2 Our Moving Object Segmentation Model

In our case, taking into consideration the motion information obtained by calculating the optical flow, we propose the following descriptors for the segmentation of mobile objects in a video surveillance dataset:

$$\begin{aligned} k_{in}(x, \Omega_{in}) &= \lambda |SV_g(x) - c_1(\Omega_{in})|^2 \\ k_{out}(x, \Omega_{out}) &= \lambda |SV_g(x) - c_2(\Omega_{out})|^2 \\ k_b(x) &= \mu \end{aligned} \tag{3}$$

With c_1 is the average of the region Ω_{in} , c_2 is the average of the region Ω_{out} , μ and λ constants positive. SVg(x) is the image obtained after a threshold of the

optical flow velocity and applicate of a gaussian filter (Figure 3). The values of c_1 and c_2 are re-estimated during the spread of the curve. The method of levels sets is used directly representing the curve $\Gamma(x)$ as the curve of zero to a continuous function $U(x)$. Regions and contour are expressed as follows:

$$\begin{aligned} \Gamma &= \partial\Omega_{in} = \{x \in \Omega_I / U(x) = 0\} \\ \Omega_{in} &= \{x \in \Omega_I / U(x) < 0\} \\ \Omega_{out} &= \{x \in \Omega_I / U(x) > 0\} \end{aligned} \tag{4}$$

The unknown sought minimizing the criterion becomes the function U . We introduce also the Heaviside function H and the measure of Dirac δ_0 defined by:

$$H(z) = \begin{cases} 1 & \text{if } z \leq 0 \\ 0 & \text{if } z > 0 \end{cases} \quad \text{et } \delta_0(z) = \frac{d}{dz}H(z)$$



Fig. 3. SV_g Image example

The criterion is then expressed through the functions U , H and δ in the following manner:

$$\begin{aligned} J(U, c_1, c_2) &= \int_{\Omega_I} \lambda |SV_g(x) - c_1|^2 H(U(x)) dx + \\ &\int_{\Omega_I} \lambda |SV_g(x) - c_2|^2 (1 - H(U(x))) dx + \\ &\int_{\Omega_I} \mu \delta(U(x)) |\nabla U(x)| dx \end{aligned} \tag{5}$$

with:

$$\begin{aligned} c_1 &= \frac{\int_{\Omega} SV_g(x) H(U(x)) dx}{\int_{\Omega} H(U(x)) dx} \\ c_2 &= \frac{\int_{\Omega} SV_g(x) (1 - H(U(x))) dx}{\int_{\Omega} (1 - H(U(x))) dx} \end{aligned} \tag{6}$$

To calculate the Euler-Lagrange equation for unknown function U , we consider a regularized versions for the functions H and δ noted H_ϵ and δ_ϵ . The evolution equation is found then expressed directly with U , the function of the level set:

$$\begin{aligned} \frac{\partial U}{\partial \tau} &= \delta_\epsilon(U) [\mu \text{div}(\frac{\nabla U}{|\nabla U|}) + \lambda |SV_g(x) - c_1|^2 \\ &- \lambda |SV_g(x) - c_2|^2] (in \Omega_I) \end{aligned} \tag{7}$$

$$\frac{\delta_\epsilon(U)}{|\nabla U|} \frac{\partial U}{\partial N} = 0 \text{ (on } \partial\Omega_I)$$

with $\text{div}(\frac{\nabla U}{|\nabla U|})$ the curvature of the level curve of U via x and $\frac{\partial U}{\partial N}$ the derivative of U compared to normal inside the curve N .

3.3 Supported Video Surveillance Events

Until now, our system supports the following 5 events:

- C1: Approaching vehicle to the camera (figure 4.a)
- C2: One or more moving vehicle (figure 4.b)
- C3: Approaching pedestrian (figure 4.c)
- C4: One or more moving pedestrian (figure 4.d)
- C5: Combined Concept (figure 4.e)

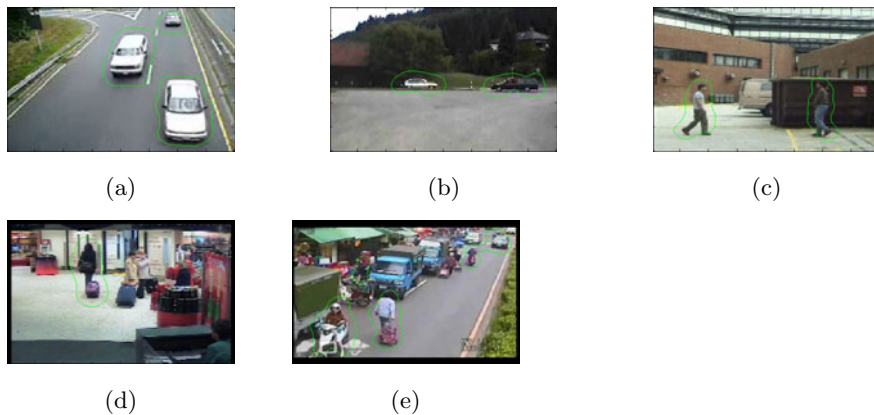


Fig. 4. Examples of images extracted from our video surveillance dataset

4 Combining Single SVM Classifier for Learning Video Event

Support Vector Machines (SVMs) have been applied successfully to solve many problems of classification and regression. However, SVMs suffer from a phenomenon called 'catastrophic forgetting', which involves loss of information learned in the presence of new training data. Learn++ [14] has recently been introduced as an incremental learning algorithm. The strength of Learn++ is its ability to learn new data without forgetting prior knowledge and without requiring access to any data already seen, even if new data introduce new classes. To benefit from the speed of SVMs and the ability of incremental learning of Learn++, we propose to use a set of trained classifiers with SVMs based on Learn++ inspired from [13]. Experimental results of detection of events suggest that the proposed combination is promising. According to the data, the performance of SVMs is similar or even superior to that of a neural network or a Gaussian mixture model.

4.1 SVM Classifier

Support Vector Machines (SVMs) are a set of supervised learning techniques to solve problems of discrimination and regression. The SVM is a generalization of linear classifiers. The SVMs have been applied to many fields (bio-informatics, information retrieval, computer vision, finance ...).

According to the data, the performance of SVMs is similar or even superior to that of a neural network or a Gaussian mixture model. They directly implement the principle of structural risk minimization [15] and work by mapping the training points into a high dimensional feature space, where a separating hyperplane (w, b) is found by maximizing the distance from the closest data points (boundary-optimization). Given a set of training samples $S = \{(x_i, y_i) | i = 1, \dots, m\}$, where $x_i \in R_n$ are input patterns, $y_i \in +1, -1$ are class labels for a 2-class problem, SVMs attempt to find a classifier $h(x)$, which minimizes the expected misclassification rate. A linear classifier $h(x)$ is a hyperplane, and can be represented as $h(x) = \text{sign}(w^T x + b)$. The optimal SVM classifier can then be found by solving a convex quadratic optimization problem:

$$\begin{aligned} \underset{w, b}{\max} \quad & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \xi_i \quad \text{subject to} \\ & y_i (\langle w, x_i \rangle + b) \geq 1 - \xi_i \quad \text{and} \quad \xi_i \geq 0 \end{aligned} \tag{8}$$

Where b is the bias, w is weight vector, and C is the regularization parameter, used to balance the classifier’s complexity and classification accuracy on the training set S . Simply replacing the involved vector inner-product with a non-linear kernel function converts linear SVM into a more flexible non-linear classifier, which is the essence of the famous kernel trick. In this case, the quadratic problem is generally solved through its dual formulation:

$$\begin{aligned} L(w, b, \alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} (\sum_{i=1}^m y_i y_j \alpha_i \alpha_j K(x_i, x_j)) \\ \text{subject to } C \geq \alpha_i \geq 0 \text{ and } \sum_{i=1}^m y_i \alpha_i y_i = 0 \end{aligned} \tag{9}$$

where α_i are the coefficients that are maximized by Lagrangian. For training samples x_i , for which the functional margin is one (and hence lie closest to the hyperplane), $\alpha_i > 0$. Only these instances are involved in the weight vector, and hence are called the support vectors [12]. The non-linear SVM classification function (optimum separating hyperplane) is then formulated in terms of these kernels as:

$$h(x) = \text{sign} \left(\sum_{i=1}^m \alpha_i y_i K(x_i, x_j) - b \right) \tag{10}$$

4.2 M-SVM Classifiers

M-SVM is based on Learn++ algorithm. This latter, generates a number of weak classifiers from a data set with known label. Depending on the errors of the classifier generated low, the algorithm modifies the distribution of elements in the subset according to strengthen the presence of the most difficult to classify. This procedure is then repeated with a different set of data from the same dataset and new classifiers are generated. By combining their outputs according to the scheme of majority voting Littlestone we obtain the final classification rule.

The weak classifiers are classifiers that provide a rough estimate - about 50% or more correct classification - a rule of decision because they must be very quick to generate. A strong classifier from the majority of his time training to refine his decision criteria. Finding a weak classifier is not a trivial problem

and the complexity of the task increases with the number of different classes, however, the use of NN algorithms can correctly resolved effectively circumvent the problem. The error is calculated by the equation:

$$error_t = \sum_{i:h_t(x_i) \neq y_i} S_t(i) [|h_t(x_i) - y_i|] \tag{11}$$

with $h_t : X \rightarrow Y$ an hypothesis and where TR_t is the subset of training subset and the TE_t is the test subset. The synaptic coefficients are updated using the following equation:

$$w_{t+1}(i) = w_t(i) * \begin{cases} \beta_t & \text{if } H_t(x_i) = y_i \\ 1 & \text{else} \end{cases} \tag{12}$$

Where t is the iteration number, B_t composite error and standard composite hypothesis H_t .

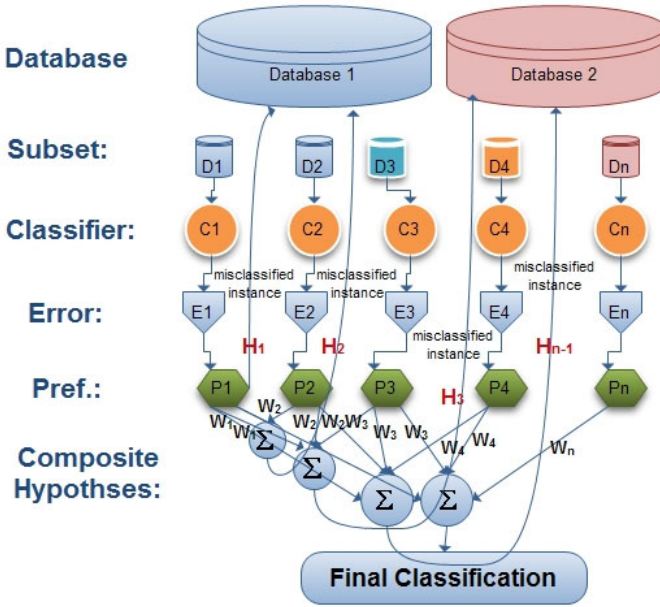


Fig. 5. M-SVM classifier

In our approach we replace each weak classifier by SVM. After T_k classifiers are generated for each D_k , the final ensemble of SVMs is obtained by the weighted majority of all composite SVMs:

$$H_{final}(x) = arg \max_{y \in Y} \sum_{k=1}^K \sum_{t:h_t(x)=y} \log \frac{1}{\beta_t} \tag{13}$$

5 Visual Feature Extraction

We use a set of different visual descriptors at various granularities for each frame, rid of the static background, of the video shots. The relative performance of the

specific features within a given feature modality is shown to be consistent across all concepts/events. However, the relative importance of one feature modality vs. another may change from one concept/event to the other. The following descriptors had the top overall performance for both search and concept modeling experiments:

- Color Histogram: global color represented as 128-dimensional histogram in HSV color space.
- Color Moments: localized color extracted from 3x3 grid and represented by the first 3 moments for each grid region in Lab color space as normalized 255-dimensional vector.
- Co-occurrence Texture: global texture represented as a normalized 96-dimensional vector of entropy, energy, contrast and homogeneity extracted from the image gray-scale co-occurrence matrix at 24 orientation.
- Gabor Texture: Gabor functions are Gaussians modulated by complex sinusoids. The Gabor filter masks can be considered as orientation and scale-tunable and line detectors. The statistics of these micro-features in a given region can be used to characterize the underlying texture information. We take 4 scales and 6 orientations of Gabor textures and further use their mean and standard deviation to represent the whole frame and result in 48 textures.
- Fourier: Features based on the Fourier transform of the binarized edge image. The 2-dimensional amplitude spectrum is smoothed and down-sampled to form a feature vector of 512 parameters.
- Sift: The SIFT descriptor [7] is consistently among the best performing interest region descriptors. SIFT describes the local shape of the interest region using edge histograms. To make the descriptor invariant, while retaining some positional information, the interest region is divided into a 4x4 grid and every sector has its own edge direction histogram (8 bins). The grid is aligned with the dominant direction of the edges in the interest region to make the descriptor rotation invariant.
- Combined Sift and Gabor.
- Wavelet Transform for texture descriptor: Wavelets are hybrids that are waves within a region of the image, but otherwise particles. Another important distinction is between particles that have place tokens and those that do not. Although all particles have places in the image, it does not follow these places will be represented by tokens in feature space. It is entirely feasible to describe some images as a set of particles, of unknown position. Something like this happens in many description of texture. We performe 3 levels of a Daubechies wavelet [4] decomposition for each frame and calculate the energy level for each scale, which resulted in 10 bins features data.
- Hough Transform: As descriptor of shape we employ a histogram based on the calculation of Hough transform [9]. This histogram gives information better than those given by the edge histogram. We obtain a combination of behavior of the pixels in the image along the straight lines.
- Motion Activity: We use the information calculated by the optical flow, through concentrating on movements of the various objects (people or

vehicle) detected by the method described in the previous section. The descriptors that we use are correspond to the energie calculated on every sub-band, by a decomposition in wavelet of the optical flow estimated between every image of the sequence. We obtain a vector of 10 bins, they represent for every image a measure of activity sensitive to the amplitude, the scale and the orientation of the movements in the shot.

6 Experimental Results

Experiments are conducted on the many sequence from TRECVID2009 database of video surveillance and many other sequences from road traffics. About 20 hours are used to train the feature extraction system, that are segmented in the shots. These shots were annotated with items in a list of 5 events. We use about 20 hours for the evaluation purpose. To evaluate the performance of our system we use the common measure from the information retrieval community: the Average Precision. Figure 6 shows the evaluation of returned shots. The best results are obtained for all events.

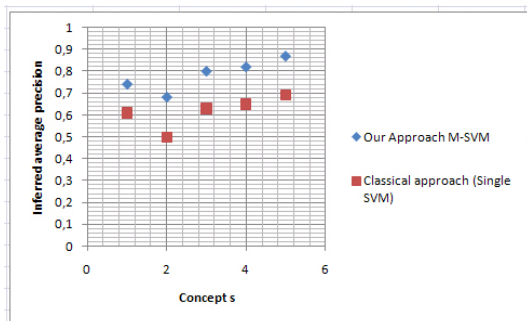


Fig. 6. Our run score versus Classical System (Single SVM) by Event

7 Conclusion

In this paper, we have presented preliminary results and experiments for high-level feature extraction for video surveillance indexing and retrieval. The results obtained so far are interesting and promoters. The advantage of this approach is that allows human operators to use context-based queries and the response to these queries is much faster. The meta-data layer allows the extraction of the motion and objects descriptors to XML files that then can be used by external applications. Finally, the system functionalities will be enhanced by a complementary tools to improve the basic concepts and events taken care of by our system.

Acknowledgement

The authors would like to acknowledge the financial support of this work by grants from General Direction of Scientific Research (DGRST), Tunisia, under the ARUB program.

References

1. Horn, B.K.P., Schunk, B.G.: Determining optical flow. *Artificial Intelligence* 17, 185–201 (1981)
2. Chan, T., Vese, L.: An active contour model without edges. In: Nielsen, M., Johansen, P., Fogh Olsen, O., Weickert, J. (eds.) *Scale-Space 1999*. LNCS, vol. 1682, p. 141. Springer, Heidelberg (1999)
3. van Liempt, M., Koelma, D.C., et al.: The mediamill trecvid 2006 semantic video search engine. In: *Proceedings of the 4th TRECVID Workshop, Gaithersburg, USA (November 2006)*
4. Daubechies, I.: CBMS-NSF series in app. Math., In: SIAM (1991)
5. Ellouze, M., Karray, H., Alimi, M.A.: Genetic algorithm for summarizing news stories. In: *Proceedings of International Conference on Computer Vision Theory and Applications, Spain, Barcelona, pp. 303–308 (March 2006)*
6. Wali, A., Karray, H., Alimi, M.A.: Sirpvct: System of indexing and the search for video plans by the contents text. In: *Proc. Treatment and Analyzes information: Methods and Applications, TAIMA 2007, Tunisia, Hammamet, pp. 291–297 (May 2007)*
7. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 2(60), 91–110 (2004)
8. Ben Aoun, N., El'Arbi, M., Ben Amar, C.: Multiresolution motion estimation and compensation for video coding. In: *10th IEEE International Conference on Signal Processing, Beijing, China (October 2010)*
9. Boujemaana, N., Ferencatu, M., Gouet, V.: Approximate search vs. precise search by visual content in cultural heritage image databases. In: *Proc. of the 4-th International Workshop on Multimedia Information Retrieval (MIR 2002) in conjunction with ACM Multimedia (2002)*
10. Karray, H., Ellouze, M., Alimi, M.A.: Using text transcriptions for summarizing arabic news video. In: *Proc. Information and Communication Technologies International Symposium, ICTIS 2007, Morocco, Fes, pp. 324–328 (April 2007)*
11. Macan, T., Loncaric, S.: Hybrid optical flow and segmentation technique for lv motion detection. In: *Proceedings of SPIE Medical Imaging, San Diego, USA, pp. 475–482 (2001)*
12. Shawe-Taylor, J., Cristianini, N.: *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, Cambridge (2000)
13. Zeki, E., Robi, P., et al.: Ensemble of svms for incremental learning. In: Oza, N.C., Polikar, R., Kittler, J., Roli, F. (eds.) *MCS 2005*. LNCS, vol. 3541, pp. 246–256. Springer, Heidelberg (2005)
14. Polikar, R., Udpa, L., Udpa, S.S., Honavar, V.: Learn++: An incremental learning algorithm for supervised neural networks. *IEEE Trans. Sys. Man, Cybernetics (C)* 31(4), 497–508 (2001)
15. Vapnik, V.: *Statistical Learning Theory* (1998)
16. Wali, A., Alimi, A.M.: Event detection from video surveillance data based on optical flow histogram and high-level feature extraction. In: *IEEE DEXA Workshops 2009, pp. 221–225 (2009)*
17. Chang, S., Jiang, W., Yanagawa, A., Zavesky, E.: Columbia university trecvid2007: High-level feature extraction. In: *Proceedings TREC Video Retrieval Evaluation Online, TRECVID 2007 (2007)*

Evaluation of Human Detection Algorithms in Image Sequences

Yannick Benezeth¹, Baptiste Hemery², H el ene Laurent³,
Bruno Emile⁴, and Christophe Rosenberger²

¹ Orange Labs, 4 rue du Clos Courtel,
35510 Cesson-S evign e, France

² Laboratoire GREYC, ENSICAEN,
- Universit e de Caen, - CNRS,

6 bd du Mar echal Juin,
14000 Caen, France

³ ENSI de Bourges,

- Institut PRISME,

88 bd Lahitolle, 18020 Bourges Cedex,

⁴ Institut PRISME, Universit e d'Orl eans,

2 av F. Mitterrand, 36000 Ch ateauroux, France

Abstract. This paper deals with the general evaluation of human detection algorithms. We first present the algorithms implemented within the *CAPTHOM* project dedicated to the development of a vision-based system for human detection and tracking in an indoor environment using a static camera. We then show how a global evaluation metric we developed for the evaluation of understanding algorithms taking into account both localization and recognition precision of each single interpretation result, can be a useful tool for industrials to guide them in the elaboration of suitable and optimized algorithms.

Keywords: Human detection, Background subtraction, Tracking, Classification, Evaluation metric, Object localization, Object recognition.

1 Introduction

Face to the huge development of image interpretation algorithm dedicated to various applications [1,2,3], such as target detection and recognition or video surveillance to name a few, the need of adapted evaluation metrics, which could help in a development of well thought-out algorithm or in the quantification of the relative performances of different algorithms, has become crucial. Wide annotated databases and metrics have been defined within several research competitions such as the Pascal VOC Challenge [4] or the French Robin Project [5] in order to evaluate object detection and recognition algorithms. Whatever these metrics either focus on the localization aspect or the recognition one, but not both together. Moreover, concerning the recognition objective, most of the competitions use Precision/Recall and ROC curves [4,6,7], evaluating the algorithms

on the whole database. An interpretation evaluation metric, taking into account both aspects and working on a single interpretation result, is then needed.

This article presents our works concerning the development of vision-based systems for human detection and tracking in a known environment using a static camera and the definition of an adaptable performance measure able to simultaneously evaluate the localization, the recognition and the detection of interpreted objects in a real scene using a manually made ground truth. If in a general way, the localization and the recognition have to be as precise as possible, the relative importance of these two aspects can change depending of the foreseen application. We describe in section 2 the successive algorithms implemented for the *CAPTHOM* project which more particularly focused on indoor environments. The proposed evaluation metric of a general image interpretation result is presented in section 3. Its potential interest is illustrated in section 4 on the *CAPTHOM* project. Section 5 presents conclusions and perspectives of this study.

2 Visual-Based System Developments for Human Detection in Image Sequences

Within the *CAPTHOM* project, we attempt to develop a human detection system to limit power consumption of buildings and to monitor low mobility persons. This project belongs to the numerous applications of human detection systems for home automation, video surveillance, etc. The foreseen system must be easily tunable and embeddable, providing an optimal compromise between false detection rate and algorithmic complexity.

The development of a reliable human detection system in videos deals with general object detection difficulties (background complexity, illumination conditions etc.) and with other specific constraints involved with human detection (high variability in skin color, weight and clothes, presence of partial occlusions, highly articulated body resulting in various appearances etc.). Despite of these difficulties, some very promising systems have already been proposed in the literature. It is especially the case of the method proposed by Viola and Jones [8] which attempts to detect humans in still images using a well-suited representation of human shapes and a classification method. We first of all implemented this method in a sliding window framework analyzing every image and using several classifiers. This method is based on Haar-like filters and adaboost. In an indoor environment, partial occlusions are actually frequent. The upper part of the body (head and shoulders) is often the only visible part. As it is clearly insufficient to seek in the image only forms similar to the human body in its whole, we implemented four classifiers: the whole body, the upper-body (front/back view), the upper-body (left view) and the upper-body (right view). In a practical way, the classifier analyzes the image with a constant shift in the horizontal and vertical direction. As the size of the person potentially present is not known a priori and the classifier has a fixed size, the image is analyzed several times by modifying the scale. The size of the image is divided by a scale factor (sf) between two scales. This method is called *Viola* [8] in the following paragraphs.



Fig. 1. Illustration of tracking result with a partial occlusion. First row: input images with interest points associated with each object, second row: tracking result.

In order to reduce the search space of classifiers localizing regions of interest in the image, we added a change detection step based on background subtraction. We chose to model each pixel in the background by a single Gaussian distribution. The detection process is then achieved through a simple probability density function thresholding. This simple model presents a good compromise between detection quality, computation time and memory requirements [9,10]. The background model is updated at three different levels: the pixel level updating each pixel with a temporal filter allowing to consider long time variations of the background, the image level to deal with global and sudden variations and the object level to deal with the entrance or the removal of static objects. This method is called *Viola* [8]+*BS* afterwards.

We finally developed a method using additionally temporal information. We propose a method using advantages of tools classically dedicated to object detection in still images in a video analysis framework. We use video analysis to interpret the content of a scene without any assumption while objects nature is determined by statistical tools derived from object detection in images. We first use background subtraction to detect objects of interest. As each connected component detected potentially corresponds to one person, each blob is independently tracked. Each tracked object is characterized by a set of points of interest. These points are tracked, frame by frame. The position of these points, regarding connected components, enables to match tracked objects with detected blobs. The tracking of points of interest is carried out with the pyramidal implementation of the Lucas and Kanade tracker [11,12]. The nature of these tracked objects is then determined using the previously described object recognition method in the video analysis framework. Figure 1 presents an example of tracking result with partial occlusion. This method is called *CAPTHOM* in the following.

For more information about the three considered methods, the interested reader can refer to [13].

3 Evaluation Metric

The developed evaluation metric [14] is based on four steps corresponding to: (i) Objects matching, (ii) Local evaluation of each matched object in terms of localization and recognition, (iii) Over- and under-detection compensation and (iv) Global evaluation score computation of the considered interpretation result.

Figure 2 illustrates the different stages on an original image extracted from the 2007 Pascal VOC challenge. For this image, the ground truth is composed of 4 objects which all belong to the human class. The interpretation result contains as for it two detected persons. We can note that the first person of the ground truth is well localized and recognized. The last three persons are well recognized but poorly localized. Indeed, only one object has been detected instead of three.

The first step, consisting in matching the objects of the ground truth and of the interpretation result, is done using the *PAS* metric [4]:

$$PAS(I_{gt}, I_i, u, v) = \frac{\text{Card}(I_{gt}^{r(u)} \cap I_i^{r(v)})}{\text{Card}(I_{gt}^{r(u)} \cup I_i^{r(v)})} \quad (1)$$

with $\text{card}(I_{gt}^{r(u)})$ the number of pixels from the object u in the ground truth, and $\text{card}(I_i^{r(v)})$ the number of pixels from the detected object v in the interpretation result. The number of rows of the resulting matching score matrix corresponds to the number of objects in the ground truth, and the number of columns corresponds to the number of objects in the interpretation result. This matrix is computed, as in [15]. The values range from 0 to 1, 1 corresponding to a perfect localization. From the matching score matrix, we can match objects by two methods: the first one consists in using an Hungarian algorithm, which implies one-to-one matching as in [4]; the second one consists in simply applying a threshold, which enables multiple detections as in [16]. We use the threshold method, with a threshold set to 0.2 by default, as it allows that each object of the interpretation result can be assigned to several objects from the ground truth or vice-versa. The first person of the ground truth (object 1) is well localized in the interpretation result (object 2). Their recovery score exceeding the threshold, they are matched resulting in value 1 in the corresponding cell of the assignment matrix. Concerning the persons group, only two objects of the ground truth (objects 3 and 4) are matched with the one object of the interpretation result (object 1).

The second step consists in the local interpretation evaluation of each matched object. The localization is first evaluated using the *Martin* metric [17] adapted to one object:

$$S_{loc}(I_{gt}, I_i, u, v) = \min \left(\frac{\text{card}(I_{gt}^{r(u)} \setminus I_i)}{\text{card}(I_{gt}^{r(u)})}, \frac{\text{card}(I_i^{r(v)} \setminus I_{gt})}{\text{card}(I_i^{r(v)})} \right) \quad (2)$$

with $\text{card}(I_{gt}^{r(u)})$ the number of pixels of object u present in the ground truth and $\text{card}(I_{gt}^{r(u)} \setminus I_i)$ the number of pixels of object u present in the ground truth but not

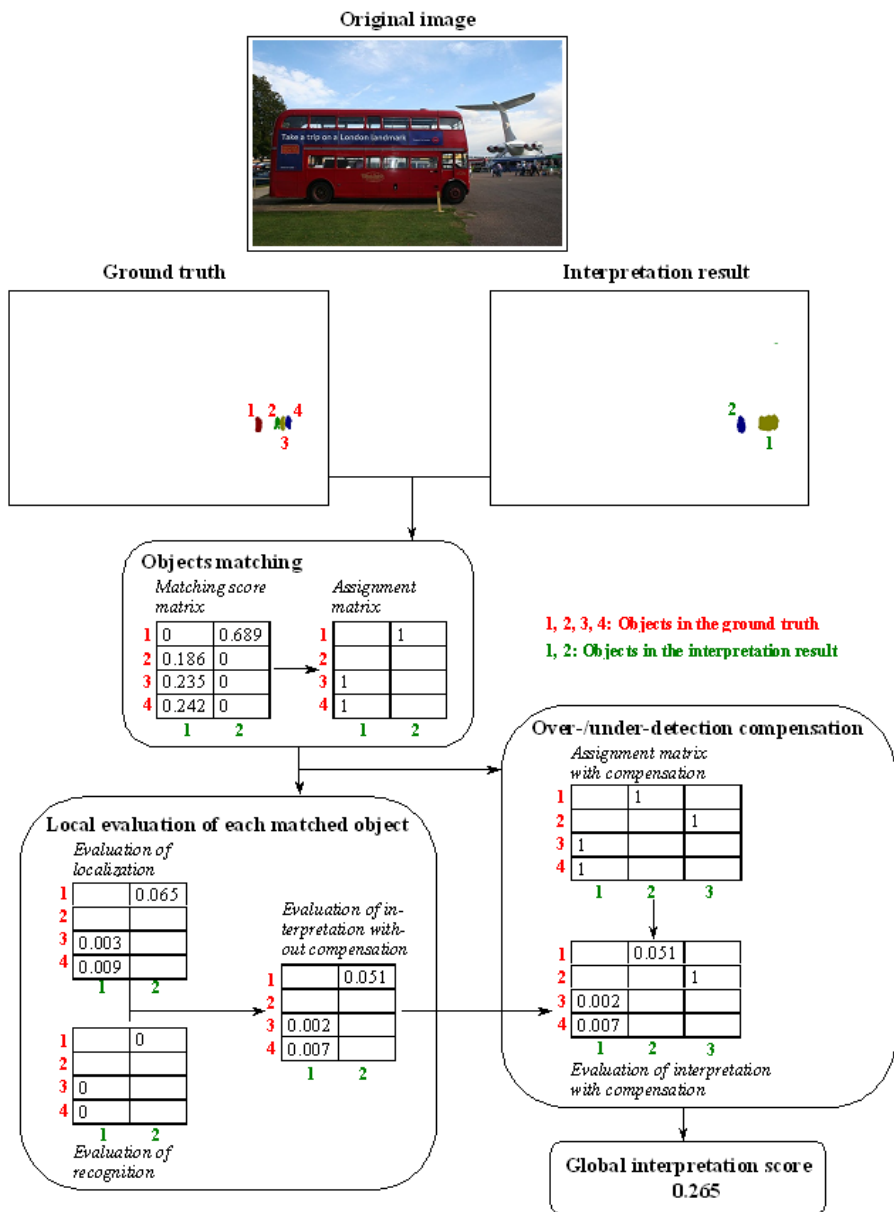


Fig. 2. Illustration of the global evaluation of an interpretation result

present in the interpretation result. This metric has been chosen according to the comparative study conducted in [18] on the performances of 33 localization metrics face to different alterations like translation, scale change, rotation... The obtained localization score ranges from 0 to 1, 0 corresponding to a perfect recovery between the two objects and consequently to a perfect localization. We can note that all the matched objects are quite well localized obtaining low scores, the poorest score 0.065 corresponding to the second object of the interpretation result, namely the lonely person. The evaluation of the recognition part consists in comparing the class of the object in the ground truth and in the interpretation result. This comparison can be done in different ways. A distance matrix between each class present in the database can be for example provided, which would enable to precisely evaluate recognition mistakes. On an other way, numerous real systems track one specific class of objects and do not tolerate some approximation in the recognition step. They work in an all or nothing scheme. $S_{rec}(I_{gt}, I_i, u, v) = 0$ if classes are the same and 1 otherwise. It is the case in the developed human detection system where all detections correspond *de facto* to the right class, namely a human. The recognition evaluation matrix containing only ones, the misclassification is then indirectly highly penalized through the over and under-detection compensation. As we have to maintain an important weight for the penalization of bad localization, we choose a high value of the α parameter ($\alpha = 0.8$). We finally compute the local interpretation score $S(u, v)$ between two matched objects as a combination of the localization and the recognition scores:

$$S(u, v) = \alpha * S_{loc}(I_{gt}, I_i, u, v) + (1 - \alpha) * S_{rec}(I_{gt}, I_i, u, v) \quad (3)$$

The third step is the compensation one. Working on the assignment matrix, empty rows or columns are tracked and completed. In our example, there is no empty column meaning that all objects of the interpretation result have been matched with at least one object of the ground truth. There is consequently no over-detection. On the other hand, one row (2) is empty; one object of the ground truth has not been detected. This under-detection is compensated adding one column with score 1 at the corresponding line.

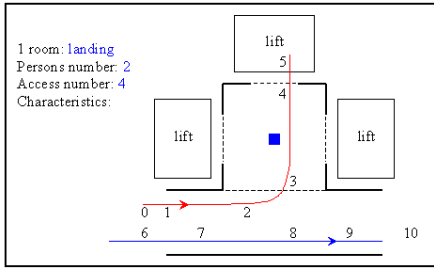
Finally, the global interpretation score is computed, taking into account the compensation stage and averaging the local interpretation scores.

4 Evaluation of Human Detection Algorithms

In order to evaluate the detection methods presented in section 2, we realized a set of reference scenarios corresponding to the specific needs expressed by the industrial partners involved in the CAPTHOM project. An extract of a scenario example is presented in figure 3. At each location, a set of characteristics (temperature, speed, posture, activity...) is associated with the formalism defined within the CAPTHOM project [19].

The three classes of scenarios from which we have built the evaluation dataset are:

Scenario No. 4 (Lift landing, entrance hall, coffee machine)



Non enclature

- | | | |
|----------------------------|------------------|----------------------------------|
| <i>Temperature:</i> | <i>Activity:</i> | <i>False detection stimulus:</i> |
| 1: $T < T_{min}$ | A1: Walking | S0: No stimulus |
| 2: $T_{min} < T < T_{max}$ | A2: Still | S1: Close aperture |
| 3: $T > T_{max}$ | A3: Sleeping | S2: Pet |
| | A4: Reading | S3: Flashlight |
| <i>Speed:</i> | A5: Eating | S4: Mobile toy |
| 1: $v < v_{min}$ | A6: Running | S5: |
| 2: $v_{min} < v < v_{max}$ | A7: | |
| 3: $v > v_{max}$ | | <i>Interfering flow:</i> |
| | | F0: No interfering flow |
| <i>Posture:</i> | | F1: Flashlight |
| P1: Standing | | F2: Hot draught |
| P2: Sitting | | F3: electromagnetic wave |
| P3: Lying | | F4: |
| P4: | | |

Fig. 3. Extract of a scenario example defined by the industrial partners involved in the CAPTHOM project

- Set 1: scenarios involving a normal use of a room. In these scenarios, we need to detect humans that are static or moving, sitting or standing in offices, meeting rooms, corridors and dining rooms.
- Set 2: scenarios of unusual activities (slow or fast falls, abnormal agitation).
- Set 3: scenarios gathering all false detections stimuli (illumination variation, moving objects etc).

In the following, Set 4 is defined as the union of these 3 sets. In total, we used 29 images sequences in 10 different places. Images have a resolution of 320 x 240 and have an "average" quality. Each images sequence lasts from 2 to 10 minutes.

Figures 4 and 5 present results obtained with the CAPTHOM algorithm on videos extracted from our test dataset.

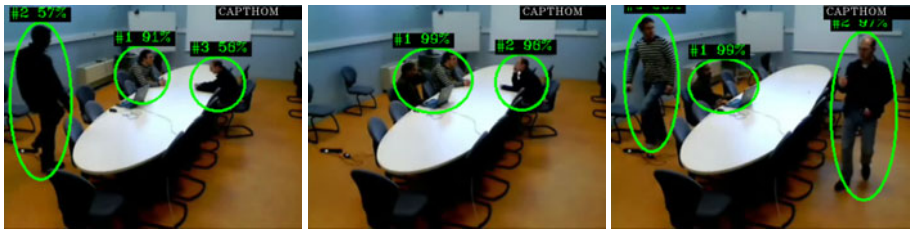


Fig. 4. Example of results obtained with the CAPTHOM method on a video presenting partial occlusion

The choice of the evaluation metric parameters, done for this study, corresponds to an expected interpretation compromise which can be encountered in many real applications. We use a parameter α , set at 0.8, to balance the localization and the recognition scores. This high value has been chosen to maintain an important weight for the penalization of bad localization. It results from a wide subjective evaluation of interpretation results we conducted, involving researchers of the French community, to better understand when a bad localization



Fig. 5. Example of results obtained with the *CAPTHOM* method on a video presenting illumination changes

is more penalizing than a misclassification [20]. One objective of this study was to be able to guide the users in the metric parameters choice and more specifically in the α ponderation parameter choice. In order to reach this objective, we asked many individuals to compare several image understanding results. We then compare the obtained subjective comparison with the objective one given by the proposed metric. With $\alpha = 0.8$, the obtained similarity rate of correct comparison was 83.33%, which shows that our metric is able to order image understanding results correctly in most of cases. Preserving good performances concerning the localization aspect will allow our system to achieve higher level information such as path or activity estimation.

Table 1 presents the mean evaluation results obtained for the three methods on the various sets of the test database using the designed interpretation evaluation metric. sf corresponds to the scale factor used from the sliding window framework analysis. We can note that the introduction of background subtraction results in algorithms that are less sensitive to the choice of this parameter. Combining properly defined test databases and an tunable evaluation metric allow the industrials to obtain a deep insight into their research developments. They can indeed quantify the performances gap between different algorithms and motivate their further technological choices. The proposed evaluation metric is also suitable for the choice of the algorithms parameters.

Table 1. Performances evaluation of the different interpretation algorithms developed within the CAPTHOM project

	Set 1	Set 2	Set 3	Set 4
Viola [8], $sf=1.1$	0.614	0.672	0.565	0.597
Viola [8], $sf=1.4$	0.719	0.707	0.105	0.436
Viola [8], $sf=1.5$	0.758	0.739	0.092	0.451
Viola [8]+BS, $sf=1.1$	0.429	0.642	0.050	0.276
Viola [8]+BS, $sf=1.4$	0.618	0.747	0.071	0.380
Viola [8]+BS, $sf=1.5$	0.663	0.745	0.082	0.405
CAPTHOM	0.338	0.089	0.043	0.176

5 Conclusion and Perspectives

We presented in this paper the potential interest of a global evaluation metric for the development of industrial understanding algorithms. The originality of the proposed measure lies in its ability to simultaneously take into account localization and recognition aspects together with the presence of over- or under-detection. Concerning the foreseen application, industrial partners involved in the project also have in mind to extend the system for car park video surveillance. In that case, the detection and distinction between different classes could be interesting and give even more sense to the misclassification error introduced in the evaluation metric. We are actually working on the use of taxonomy information for ponderating the misclassification error. The introduction of a distance matrix between classes taking into account their more or less important similarity could improve the adaptability of the proposed metric. For some applications, some misclassifications could have less repercussions than others. As an example, it could be suitable to less penalize an interpretation result where a bus is recognized as a truck, as these two objects are very similar, than an interpretation result where a bus is recognized as a building.

References

1. Cucchiara, R., Grana, C., Piccardi, M., Prati, A.: Detecting moving objects, ghosts, and shadows in video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 25(10), 1337–1342 (2003)
2. Deselaers, T., Keysers, D., Ney, H.: Improving a discriminative approach to object recognition using image patches. In: Kropatsch, W.G., Sablatnig, R., Hanbury, A. (eds.) *DAGM 2005*. LNCS, vol. 3663, pp. 326–333. Springer, Heidelberg (2005)
3. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 886–893 (2005)
4. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2008 (VOC 2008) Results, <http://www.pascalnetwork.org/challenges/VOC/voc2008/workshop/index.html>

5. D'Angelo, E., Herbin, S., Ratiéville, M.: Robin challenge evaluation principles and metrics (November 2006), <http://robin.inrialpes.fr>
6. Muller, H., Muller, W., Squire, D.M., Marchand- Maillet, S., Pun, T.: Performance evaluation in content-based image retrieval: Overview and proposals. *Pattern Recognition Letters* 22(5), 593–601 (2001)
7. Thacker, N.A., Clark, A.F., Barron, J.L., Ross Beveridge, J., Courtney, P., Crum, W.R., Ramesh, V., Clark, C.: Performance characterization in computer vision: A guide to best practices. *Computer Vision and Image Understanding* 109(3), 305–334 (2008)
8. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 511–518 (2001)
9. Wren, C., Azarbayejani, A., Darrell, T., Pentland, A.: Pfnder: Real-time tracking of the human body. *Transaction on Pattern Analysis and Machine Intelligence* (1997)
10. Benezeth, Y., Jodoin, P.M., Emile, B., Laurent, H., Rosenberger, C.: Review and Evaluation of Commonly-Implemented Background Subtraction Algorithms. In: *Proc. International Conference on Pattern Recognition, ICPR* (2008)
11. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: *Proc. International joint conference on artificial intelligence*, pp. 674–679 (1981)
12. Bouguet, J.Y.: Pyramidal implementation of the lucas kanade feature tracker: Description of the algorithm. Technical report, Intel Corporation, Microprocessor Research Labs (1999)
13. Benezeth, Y., Emile, B., Laurent, H., Rosenberger, C.: Vision-based system for human detection and tracking in indoor environment. *Special Issue on People Detection and Tracking of the International Journal of Social Robotics, IJSR* (2009)
14. Hemery, B., Laurent, H., Rosenberger, C.: Evaluation metric for image understanding. In: *Proc. IEEE International Conference on Image Processing, ICIP* (2009)
15. Phillips, I.T., Chhabra, A.K.: Empirical performance evaluation of graphics recognition systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 21(9), 849–870 (1999)
16. Wolf, C., Jolion, J.-M.: Object count/area graphs for the evaluation of object detection and segmentation algorithms. *International Journal on Document Analysis and Recognition (IJDAR)* 8(4), 280–296 (2006)
17. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: *Proc. International Conference on Computer Vision (ICCV)*, pp. 416–423 (2001)
18. Hemery, B., Laurent, H., Rosenberger, C., Emile, B.: Evaluation Protocol for Localization Metrics - Application to a Comparative Study. In: Elmoataz, A., Lezoray, O., Nouboud, F., Mammass, D. (eds.) *ICISP 2008*. LNCS, vol. 5099, pp. 273–280. Springer, Heidelberg (2008)
19. David, P., Idasiak, V., Kratz, F.: A Sensor Placement Approach for the Monitoring of Indoor Scenes. In: Kortuem, G., Finney, J., Lea, R., Sundramoorthy, V. (eds.) *EuroSSC 2007*. LNCS, vol. 4793, pp. 110–125. Springer, Heidelberg (2007)
20. Hemery, B., Laurent, H., Rosenberger, C.: Subjective Evaluation of Image Understanding Results. In: *Proc. European Signal Processing Conference, EUSIPCO* (2010)

Recognizing Objects in Smart Homes Based on Human Interaction

Chen Wu and Hamid Aghajan

AIR (Ambient Intelligence Research) Lab
Stanford University, USA
airlab.stanford.edu

Abstract. We propose a system to recognize objects with a camera network in a smart home. Recognizing objects in a home environment from images is challenging, due to the variation in object appearances such as chairs, as well as the clutters in the scene. Therefore, we propose to recognize objects through user interactions. A hierarchical activity analysis is first performed in the system to recognize fine-grained activities including eating, typing, cutting etc. The object-activity relationship is encoded in the knowledge base of a Markov logic network (MLN). MLN has the advantage of encoding relationships in an intuitive way with first-order logic syntax. It can also deal with both soft and hard constraints by associating weights to the formulas in the knowledge base. With activity observations, the defined MLN is grounded and turned into a dynamic Bayesian network (DBN) to infer object type probabilities. We expedite inference by decomposing the MLN into smaller separate domains that relates to the active activity. Experimental results are presented with our testbed smart home environment.

1 Introduction

In this paper we propose a system to recognize objects and room layout through a camera network in a smart home. Recognizing objects such as table, chair, sofa etc. in a home environment is challenging. First, many objects such as chairs and desks have varied appearances and shapes. Second, they are usually viewed from the cameras from different viewpoints. Third, Cameras installed in rooms often have a wide field of view. Images are usually cluttered with many objects while some objects of interest may have small image size. However, many objects are defined by their functions to users and not necessarily by their appearance. Such objects can be recognized indirectly from human activities during interaction with the objects.

In our work objects in the kitchen, dining room, living room and study room are recognized based on the activity analyzed from the camera network. The object types and activity classes in each semantic location are listed in Table 1. We adopt a hierarchical approach for activity recognition, including coarse- and fine-level activity recognition with different image features. In addition to the simpler pose-related activities such as standing, sitting and lying, we are also

Table 1. Activity classes used in this work and the objects recognized in each semantic location context

location	activity	objects
kitchen	walking, standing, cutting, scrambling	worktop, microwave, floor
dining room	walking, standing, sitting, eating	dining table, floor
living room	walking, standing, sitting, lying, watching	floor, chair, sofa, TV
study room	walking, standing, sitting, typing	computer, chair, floor

able to recognize activities involving subtle motions, such as cutting, scrambling, eating, typing etc. The fine-level analysis of activities enables discrimination of more types of objects in the environment.

To infer objects the relationship between objects and activities needs to be modeled. Probabilistic graphical models are good candidates for modeling the relations between objects, user activities and other events. However, such relationships can be quite complex in real applications, and building a graphical model manually can become intractable as its scale increases. Moreover, a single inclusion or removal of a variable or a modification of a relation may result in many changes in the graphical model. It is therefore crucial to employ a framework which can **(a)** handle such complex relations in an intuitive and scalable fashion, and **(b)** model the vision output and high-level deductions in a statistical way. In this paper we use Markov logic network (MLN) [1] to interface vision processing outputs and high-level reasoning. MLN can be regarded as a template to construct Markov networks. The advantage of MLN is that it intuitively models various relations between objects and user activities in first-order logic, which serves as the knowledge base for inference. Each formula in the knowledge base has a weight, representing the confidence associated with it. With observations, MLN is grounded into a Markov random field (MRF). Therefore, the probability of variables can be inferred through the MRF. MLN has been used in event recognition in visual surveillance [2] where its advantage in accommodating commonsense knowledge into event inference is demonstrated.

The contributions of this work are as follows. (1) We propose to recognize objects through human activities when the object category has changing appearance and when the object can be identified through human interaction. This approach is especially helpful for recognizing objects in a smart home environment. (2) We demonstrate that fine-level activities in the home environment can be analyzed and they are effective to differentiate many types of objects. (3) We propose to use Markov logic network to interface vision and semantic reasoning, and to encode the relational structure between objects and user activities in our prior knowledge. The model is capable of handling complex relationships in a scalable way. Another advantage of MLN over Markov networks is that it can handle both soft and hard constraints (relationships), which we exploit in our approach.

The rest of the paper is organized as follows. Sec. 2 summarizes related work on object recognition and activity classification. The overview structure of our system is presented in Sec. 3. The hierarchical activity recognition with multiple

cameras is briefly explained in Sec. 4. Sec. 5 presents the MLN knowledge base used for our problem and the inference flow and considerations. The testbed and experimental results are described in Sec. 6.

2 Related Work

Image-based object detection approaches are based on local appearance models, grouping geometric primitives and learning from image patterns [3]. Recent work based on using image contextual information indicates promising results on object detection [4,5].

Using human activity as context to detect objects relies upon modeling the relationship between activities and objects, as well as on vision-based analysis to infer the activities. In [6] Peursum et al. label image segments with objects such as floor, chair, keyboard, printer and paper in an office, based on features of human pose. Gupta et al. [7] detect manipulable objects (cup, spray bottle, phone, flashlight) from manipulation motion, reach motion, object reaction and object evidence from an appearance-based object detector. Both approaches define a Bayesian model which employs image features and action or pose features to infer the object type. Such an approach may be sensitive to the environment and placement of cameras since vision processing is dependent on such factors. But semantic reasoning of object labels is less dependent on camera views and more a function of the deduced user activities. Therefore, separating vision processing from semantic reasoning allows to transfer the latter module to other environments. Similar observation is made in [8], where layered hidden Markov models are used.

Vision-based human activity analysis has seen significant progress in recent years [9], including advances in analyzing realistic activities from videos of the public domain [10]. However, there are only a few works that focus on activity recognition in the home environment. In [11], situation models are extracted from video sequences of a smart environment, which are essentially semantic-level activities including both individual activities and two-person interactions. Both [12] and [13] use video data and RFID for activity recognition. Wu et al. in [12] use RFID readings and object detection from video to jointly estimate activity and object use. The learning process is bootstrapped with commonsense knowledge mined from the internet, and the object model from the video is automatically acquired without labeling by leveraging RFID readings. Their work infers activity indirectly from object use. Park et al. compare activity recognition with RFID and vision [13]. They conclude that for kitchen activities which involve more object usage and for which visual features (e.g., silhouettes) are not very distinguishable, RFID-based recognition has higher performance while vision-based recognition accuracy is higher for living room activities.

3 System Overview

Fig. 1(a) shows the two main steps for object recognition in our system. The first step is activity analysis in the camera network. A detailed illustration of

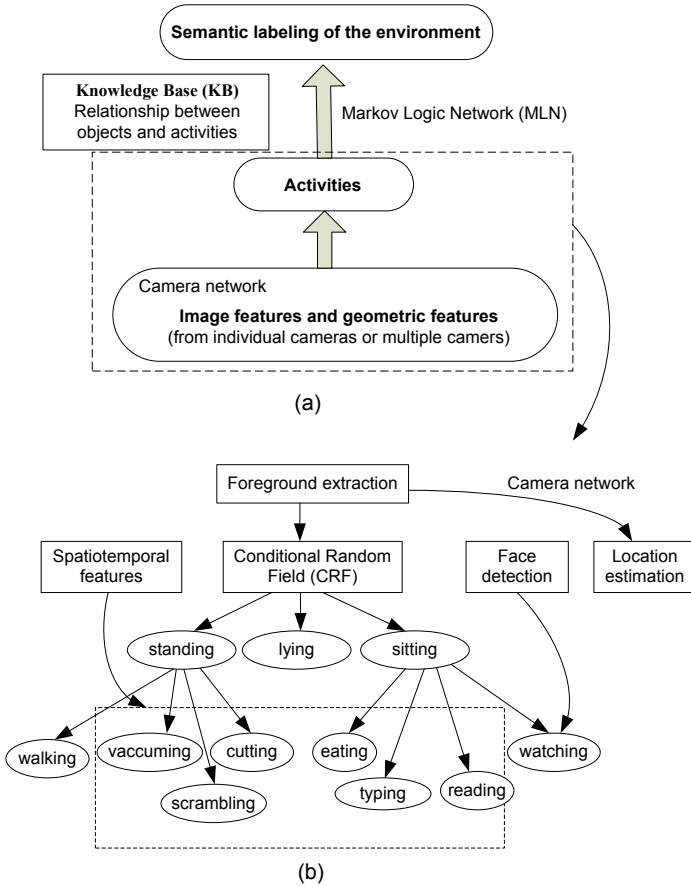


Fig. 1. Overview of system modules. (a) Layered structure for object recognition through human activities in the smart home. (b) Hierarchical activity analysis through different types of image features. The activities detected in the smart home are shown in ellipses.

the hierarchical activity analysis can be found in Fig. 1(b). This step yields the location and activity of the person. Note that not all activities shown in Fig. 1(b) are used in object recognition (Table 1) because some activities are not directly related to the environment objects.

In the second step, the room is divided into grids of size $30cm \times 30cm$ and object type of each grid is inferred with the activity observed in that grid. Object-activity relationship is defined in the knowledge base of MLN. Activity observations are converted into evidence predicates to input to the MLN model. The related MLN variables and formulas are activated and converted into MRF to infer object type probability related to the activity. Finally, each grid in the room will have a probability distribution over all object types, so that the objects are identified as grids with high probability of its type.

4 Activity Analysis in the Camera Network

The major challenges for activity recognition in the home environment include: 1. The person is often occluded by furniture; 2. Since the person freely moves and turns around while the cameras are static, the cameras may not always have a good viewpoint to observe the activity; 3. Activities in the home can have quite disparate characteristics. While activities such as lying can be distinguished from the pose, the kitchen activities usually have simple poses with subtle hand motions; 4. A fusion mechanism is needed either at the feature or the decision level.

As the whole activity recognition system, we use a hierarchical approach to classify user activities with visual analysis in a two-level process. Different types of activities are often represented by different image features, hence attempting to classify all activities with a single approach would be ineffective. In Fig. 1(b), activities are represented by coarse and fine levels. The coarse activity level includes the classes of *standing*, *sitting* and *lying*, which relate to the pose of the user. Adding global motion information and face detection, more attributes are added to *standing* and *sitting* to discriminate *walking* and *watching* in the second level. The fine activity level also consists of activities involving movement such as *cutting*, *eating*, *reading*, etc. We apply such a hierarchical approach because the first-level activities are discriminated based on pose, while the second-level activities are classified based on motion features.

In the first level, activity is coarsely classified into *standing*, *sitting* and *lying* with temporal conditional random field (CRF), through employing a set of features consisting of the height of the user (through 3D tracking) and the aspect ratio of the user's bounding box. Details of the process and performance evaluation can be found in [14].

Based on the result of the coarse level, the activity is further classified at the fine-level based on several image features. The local motion related activities are recognized based on spatio-temporal features [15]. A codebook of size N is constructed with K-means clustering on a random subset of all the extracted spatio-temporal features of the training dataset. Each feature is assigned to the closest cluster in Euclidean distance. The video sequences are segmented into episodes with duration of t seconds. Bag-of-features (BoF) are collected for every episode, therefore each episode has the histogram of spatio-temporal features as its feature vector. We use discriminative learning with SVM. Note that we also have *others* as an activity category in the experiments. This is because our sequences are not specifically designed for the defined activity types. There are many observations where the activities are in transition phase or the person is simply doing some activities at random which are not within our defined categories. This is also a challenge for our activity recognition algorithm, since due to the fact that *others* includes many different motions, the feature space for *others* is complex. However, the applications built on top of activity analysis discussed in this paper are less sensitive to false positives on *others*, because the system is usually designed to perform no operation when the user's activity is not specific. Details of the experiments and performance can be found in [16].

In the second level, some other image features are used as well. *Standing* is classified as *walking* when global motion is detected. Face detection is used when the person is *sitting*, to identify *watching* action. Motion templates (from OpenCV) is used to detect the action of *open-close* the door of microwave from the side view. The assumption is that when the person has an *open-close* action, there will be horizontal motion templates in one direction followed by those in the opposite direction. The probability of the *open-close* action is express as follows:

$$p(t) = \exp\left(-\frac{(N_1 - N)^2}{\sigma^2}\right) \exp\left(-\frac{(N_2 - N)^2}{\sigma^2}\right) \quad (1)$$

where N is the window of N frames, N_1 represents the number of frames with horizontal motion segments in one direction in the previous N frames, while N_2 represents the number of frames with motion segments in the opposite direction in the next N frames of t . σ indicates the magnitude of such regular motion patterns we expect. In our experiments, $N = 10$ and $\sigma = 6$.

5 Object Recognition from Object-Activity Relationship

The knowledge base of MLN for object recognition can be found in Knowledge Base [1](#). Markov logic network consists of a set of pairs (F_i, w_i) , e.g., formulas and their weights. The formulas defined relations between the variables as “rules”, but such rules are soft ones. Likelihood of the formulas are indicated by the probability at the beginning of the lines. Lines 2 and 3 define variables (object and activity) and their values. Lines 6-8 define the predicates used in the domain. $Hasact(act, t)$ represents the activity at the grid at a time t , while $Hastype(object, t)$ represents the grid object type at t . $After(t2, t1)$ indicates $t2$ is after $t1$. Lines 11-14 specify mutual exclusiveness of variables act and $object$. The formulas are defined in first-order logic syntax. For example, line 17 means if the previous object type of the grid is *Other* and the current observation is *Walking*, the likelihood of *Floor* is p_1 . The formula is a hard constraint when $p = 1$. For example in line 22, it means with *Sitting* observation, *Sofa* remains its identity and likelihood.

MLN defined in Knowledge Base [1](#) can be applied for all grids. One option is to ground the whole MLN with activity observations from a selected time interval. In this case the resulting model will have many random variables since each grounded predicate will be considered as a random variable. Inference on such a graphical model is time consuming. However, in our case each activity does not have relationships with each other (except for the mutual exclusiveness), and each relates to a small number of object types compared to all object types. Therefore, for each activity a_i , a minimal domain \mathbf{D}_i is constructed to infer its related object types. Fig. [2](#) shows the flow of processing. At each frame, if there is an activity detected, the associated grid position is computed. For all activities except for *Watching*, the person’s location is regarded as the grid of the activity. For *Watching*, all grids that are 1) in the gaze direction and 2) have a distance of at least d away from the person are regarded as the range of *Watching*. Then

Knowledge Base 1. MLN for object-activity relationship

```

1: // define variables and constants
2: object = {Floor, Chair, Sofa, Worktop, Microwave, Dtable, TV, Computer, Other
  }
3: act = { Walking, Standing, Sitting, Lying, Cutting, Scrambling, Eating, Typing,
  Watching, OpenClose }
4:
5: // predicate declaration
6: After(time,time)
7: Hasact(act,time)
8: Hastype(object,time)
9:
10: // formulas
11: 1 (ac!=ac')^Hasact(ac,t)⇒¬Hasact(ac',t)
12: 1 ∀t ∃ac Hasact(ac,t)
13: 1 (ob!=ob')^Hastype(ob,t)⇒¬Hastype(ob',t)
14: 1 ∀t ∃ob Hastype(ob,t)
15:
16: // floor
17: p1 Hastype(Other,t1)^Hasact(Walking,t2)^After(t2,t1)⇒Hastype(Floor,t2)
18: // sitting and lying activities
19: p2 Hastype(Other,t1)^Hasact(Sitting,t2)^After(t2,t1)⇒Hastype(Chair,t2)
20: p3 Hastype(Other,t1)^Hasact(Lying,t2)^After(t2,t1)⇒Hastype(Sofa,t2)
21: p4 Hastype(Chair,t1)^Hasact(Lying,t2)^After(t2,t1)⇒Hastype(Sofa,t2)
22: 1 Hastype(Sofa,t1)^Hasact(Sitting,t2)^After(t2,t1)⇒Hastype(Sofa,t2)
23: 1 Hastype(Dtable,t1)^Hasact(Sitting,t2)^After(t2,t1)⇒Hastype(Dtable,t2)
24:
25: // kitchen
26: p5 Hastype(Other,t1)^Hasact(Cutting,t2)^After(t2,t1)⇒Hastype(Worktop,t2)
27: p6 Hastype(Other,t1)^Hasact(Scrambling,t2)^After(t2,t1)
28:                                     ⇒Hastype(Worktop,t2)
29: p7 Hastype(Other,t1)^Hasact(OpenClose,t2)^After(t2,t1)
30:                                     ⇒Hastype(Microwave,t2)
31:
32: // dining table
33: p8 Hastype(Other,t1)^Hasact(Eating,t2)^After(t2,t1)⇒Hastype(Dtable,t2)
34: // living room
35: p9 Hastype(Other,t1)^Hasact(Watching,t2)^After(t2,t1)⇒Hastype(TV,t2)
36: // study room
37: p10 Hastype(Other,t1)^Hasact(Typing,t2)^After(t2,t1)⇒Hastype(Computer,t2)

```

Algorithm 2. Algorithm to separate domains

Input: Original domain \mathbf{D}_0
Output: A set of separate domains $\mathcal{D} = \{\mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_M\}$
for each activity a_i **do**
 Find the set of universal formulas $F_u = \{f_1^u, f_2^u, \dots\}$.
 Find the set of non-universal formulas related to a_i : $F_{nu} = \{f_1^{nu}, f_2^{nu}, \dots\}$.
 Get the object set in F_{nu} : O_i .
 Convert F_{nu} in conditional pdf form F'_{nu} .
 Form domain $\mathbf{D}_i = \{a_i, O_i, F_u, F'_{nu}\}$.
end for

for each grid associated with activity a_i , the sub-domain \mathbf{D}_i is activated. Since \mathbf{D}_i only changes probability of objects in its domain, probability of objects not in the domain is scaled to ensure all object probabilities sum up to 1.

Algorithm 2 shows the algorithm to convert the main domain \mathbf{D}_0 of Knowledge Base 1 into a set of separate domains $\mathcal{D} = \{\mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_M\}$. Universal formulas refer to those that apply to all activities or object types. In Algorithm 2 F_{nu} are converted to conditional pdf form of formulas, because we would like to ground the MLN into a Dynamic Bayesian network (DBN) which is a directed graph, so that $Prob(object, t)$ depends on activity observations before t .

6 Experiments

We conducted the experiments in a test-bed smart home environment, called the AIR (Ambient Intelligent Research) Lab. It is a smart studio located at Stanford University (Fig. 3). It consists of a living room, kitchen, dining area, and study area. The testbed is equipped with a network of cameras, a large-screen TV, a digital window (projected wall), handheld PDA devices, appliances, and wireless controllers for lights and ambient colors. Fig. 4 shows snapshots of several users engaged in different activities. Our video data involve four users. There are six scenarios in total, each captured by 5 synchronized cameras. In the scenario, one user does different activities at his/her own choice of sequence, for around 10 minutes. The activity models are trained on a different dataset described in [16].

To evaluate recognition rate, the object types are labeled on the grids and compared with inference results (Table 2). In Table 2, results are processed at the end of each scenario, and the precision shown is calculated by putting results from all scenarios together. Recall is obtained by calculating how many of the labeled grids for an object are covered correctly after inference. For each grid, the object type with the highest probability is chosen as the object type for that grid. Fig. 5 shows the room schematic overlaid on grids, with different color showing different objects. From Table 2 we can see that recall is generally lower, because we may not have enough observations that cover all possible object locations, e.g., there is a large floor area the person has not walked into. However, part of the objects are covered after recognition. Besides, there is usually a shift between the recognized object position and the real object position. This is

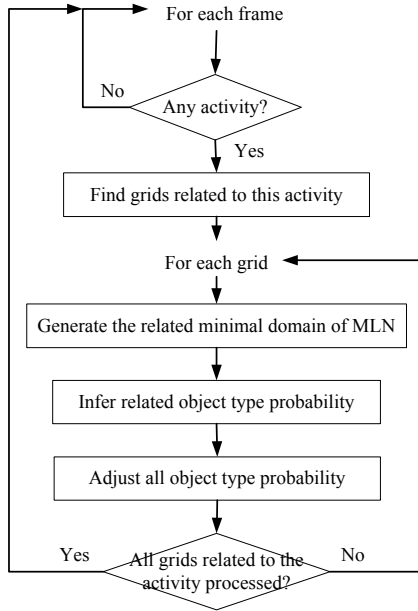


Fig. 2. The flowchart of reasoning of object types for grids

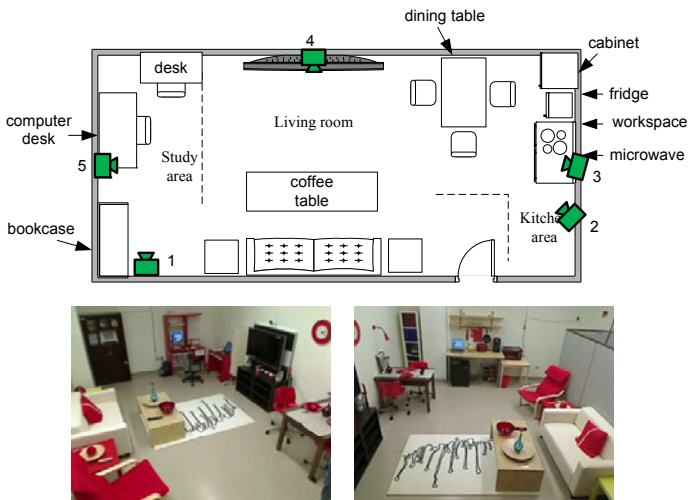


Fig. 3. The schematic and two views of AIR lab

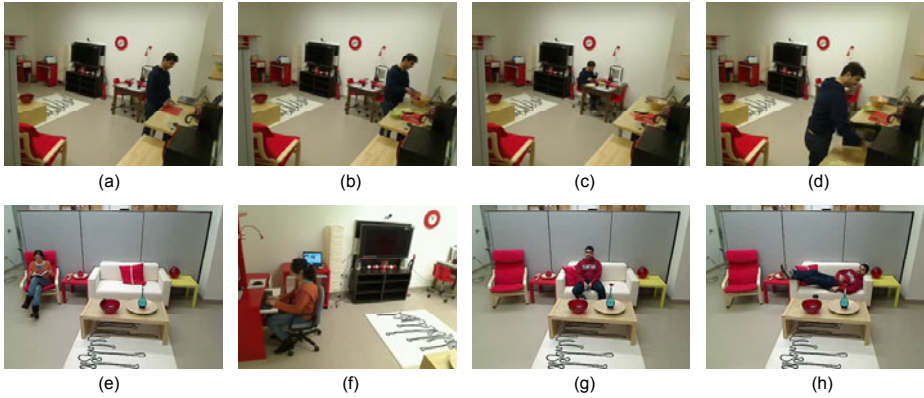


Fig. 4. Example images of some activities. (a) cutting; (b) scrambling; (c) eating; (d) using microwave; (e) reading; (f) typing on computer; (g) sitting and watching TV; (h) lying on sofa.

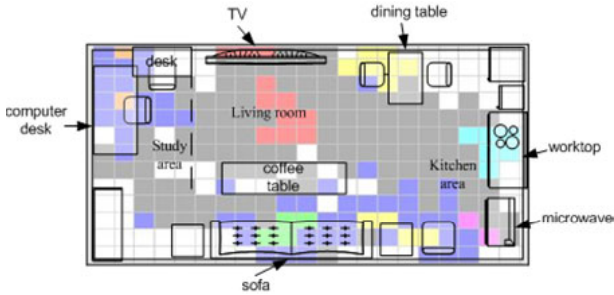


Fig. 5. The room layout and estimated grid object types shown in color: floor (gray), chair (blue), sofa (green), worktop (cyan), microwave (magenta), TV (red), Dtable (yellow), computer (orange)

Table 2. Precision and recall for AIR lab object recognition, in terms of the number of recognized grids. Note that recall* is not the recall of the algorithm, since the user’s activity does not cover all possible locations of the objects. But the recall number is nevertheless calculated by dividing the number of recognized grids with the total number of grids occupied by the object. So recall* is expectedly lower than the algorithm’s recall.

	floor	chair	sofa	worktop	microwave	Dtable	TV	computer
precision	0.88	0.64	0.8	1	1	0.78	0.27	0.5
recall*	0.47	0.82	0.5	0.67	0.5	0.67	0.75	0.17

because without further clues, we identify the person's location as the object location he/she is interacting with. However, in practice the object is usually a short distance away from the person. But still the results are helpful to indicate the objects and their location in the environment. Note that in front of the sofa, there is a region misclassified as TV. This is because the algorithm generously considers all grids along the gaze direction are possible TV locations. While some grids are identified as other objects if other activities happened there, some (like this region) have not been attended by the person. Therefore they retained the hypothesis of being TV. This also explains the low precision of TV. Further observations on this region would help resolve its identity.

7 Conclusion

In this paper we described a system to recognize objects in the smart home environment with camera network. The objects are recognized through object-activity interactions. A hierarchical activity recognition process is described, which provides fine-grained activities. The object-activity relationship is encoded in the knowledge base of MLN. We described the details of the knowledge base and inference process. Experiments are shown in the AIR lab smart home environment. Future work includes combining the position-based object type inference with image segmentation for better localization of objects.

References

1. Richardson, M., Domingos, P.: Markov logic networks. *Machine Learning* 62, 107–136 (2006)
2. Tran, S.D., Davis, L.S.: Event modeling and recognition using markov logic networks. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part II*. LNCS, vol. 5303, pp. 610–623. Springer, Heidelberg (2008)
3. Pinz, A.: *Object Categorization. Foundations and Trends in Computer Graphics and Vision* by Now Publishers (2006)
4. Oliva, A., Torralba, A.: The role of context in object recognition. *Trends in Cognitive Sciences* 11, 520–527 (2007)
5. Heitz, G., Koller, D.: Learning spatial context: Using stuff to find things. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part I*. LNCS, vol. 5302, pp. 30–43. Springer, Heidelberg (2008)
6. Peursum, P., West, G., Venkatesh, S.: Combining image regions and human activity for indirect object recognition in indoor wide-angle views. In: *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV 2005)*, vol. 1, pp. 82–89 (2005)
7. Gupta, A., Davis, L.: Objects in action: An approach for combining action understanding and object perception. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8 (2007)
8. Oliver, N., Garg, A., Horvitz, E.: Layered representations for learning and inferring office activity from multiple sensory channels. *Computer Vision and Image Understanding* 96, 163–180 (2004)

9. Moeslund, T., Hilton, A., Kruger, V.: A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding* 103, 90–126 (2006)
10. Liu, J., Luo, J., Shah, M.: Recognizing realistic actions from videos “in the wild”. In: *IEEE International Conference on Computer Vision and Pattern Recognition* (2009)
11. Brdiczka, O., Crowley, J.L., Reignier, P.: Learning situation models in a smart home. *Trans. Sys. Man Cyber. Part B* 39, 56–63 (2009)
12. Wu, J., Osuntogun, A., Choudhury, T., Philipose, M., Rehg, J.: A scalable approach to activity recognition based on object use. In: *IEEE Int. Conf. on Computer Vision*, pp. 1–8 (2007)
13. Park, S., Kautz, H.: Privacy-preserving recognition of activities in daily living from multi-view silhouettes and rfid-based training. In: *AAAI Symposium on AI in Eldercare: New Solutions to Old Problems* (2008)
14. Wu, C., Aghajan, H.: User-centric environment discovery in smart home with camera networks. *IEEE Transactions on System, Man and Cybernetics, Part A* (to appear)
15. Laptev, I.: On space-time interest points. *Int. J. Comput. Vision* 64, 107–123 (2005)
16. Khalili, A.H., Wu, C., Aghajan, H.: Hierarchical preference learning for light control from user feedback. In: *Third IEEE Workshop on CVPR for Human Communicative Behavior Analysis* (2010)

Football Players Classification in a Multi-camera Environment

Pier Luigi Mazzeo, Paolo Spagnolo, Marco Leo, and Tiziana D'Orazio

Istituto di Studi sui Sistemi Intelligenti per l'Automazione, C.N.R.

Via G. Amendola 122/D 70126 Bari, Italy

{mazzeo,dorazio,leo,spagnolo}@ba.issia.cnr.it

<http://www.issia.cnr.it/>

Abstract. In order to perform automatic analysis of sport videos acquired from a multi-sensing environment, it is fundamental to face the problem of automatic football team discrimination. A correct assignment of each player to the relative team is a preliminary task that together with player detection and tracking algorithms can strongly affect any high level semantic analysis. Supervised approaches for object classification, require the construction of ad hoc models before the processing and also a manual selection of different player patches belonging to the team classes. The idea of this paper is to collect the players patches coming from six different cameras, and after a pre-processing step based on CBTF (Cumulative Brightness Transfer Function) studying and comparing different unsupervised method for classification. The pre-processing step based on CBTF has been implemented in order to mitigate difference in appearance between images acquired by different cameras. We tested three different unsupervised classification algorithms (MBSAS - a sequential clustering algorithm; BCLS - a competitive one; and k-means - a hard-clustering algorithm) on the transformed patches. Results obtained by comparing different set of features with different classifiers are proposed. Experimental results have been carried out on different real matches of the Italian Serie A.

1 Introduction

In last years sport applications of computer vision are increasing in many contexts: in particular, many works focus on football applications, since it is one among the most popular team sports around the world, and it has a large audience in all the television programs. The research activities in sports video have focused mainly on semantic annotation [1], event detection [14] and summarization [3]. The high level applications above mentioned are based on structural low level procedures: the player segmentation [4], tracking [10] and their classification [6].

In this work we focus our attention mostly on the last aspect of image analysis: the automatic classification of players according to their team membership in a multi-camera context. Automatic team discrimination is very important because it allows to both reduce the interaction of human people and make the whole

system less dependent from particular match conditions (for example the a-priori knowledge about the team uniforms). Supervised approaches based on spectral contents are proposed in [12] (based on the analysis of colors in HSI space), [10], [13]. In [11] the position of each player in the field is integrated to make the classification more reliable. A recent interesting work working on broadcast moving images has been proposed in [2]. Moreover in a multi-view context a Cumulative Brightness Transfer Function (CBTF) is proposed [7] for mapping color between cameras located at different physical sites, which makes use of the available color information from a very sparse training set. A bi-directional mapping approach is used to obtain an accurate similarity measure between pairs of candidate objects.

All the above works try to solve the problem of player team discrimination in a supervised way and on a single camera view, by means of human-machine interactions for the creation of the reference classes. In this work we investigate on the usability of unsupervised algorithms for the automatic generation of the class models from patches coming from different cameras (players and referee). The proposed work analyzes two main aspects of unsupervised classification: the selection of the best set of features, and the selection of the best classifier for the examined application context. Moreover, the problem of different appearance of players in different views, or in differently lighted regions in the same view, is analyzed; an approach based on the evaluation of the Cumulative Brightness Transfer Function (CBTF) [5] with the goal of referring each player appearance to the same color model is proposed. Several factors, such as varying lighting conditions during the match, the overall shape similarity among players, time constraints for real time processing, make a football match a challenging arena for pattern recognition based on color descriptors. Therefore, this work try to be a starting point for all researchers that approach the problem of automatic analysis of football videos.

We started from the players segmentation algorithm proposed in [8]. For each detected player, different feature set have been tested: in particular, we have compared performance obtained with RGB histograms, rg normalized histograms and the transformed RGB (standard RGB histogram modified in order to obtain histogram with zero means and standard deviation equal to one). Then, three different unsupervised classification algorithms have been implemented and tested. We have chosen a sequential algorithm (MBSAS - Modified Basic Sequential Algorithm Scheme), a competitive one (BCLS - Basic Competitive Learning Scheme), and a hard-clustering scheme (Isodata, also known as k-means). All experiments have been performed both in absence and presence of the preprocessing based on the CBTF, finalized to mitigate different color appearance between different sources.

In the rest of the paper, firstly the system overview is summarized (section 2); then features extraction procedures (section 3) and the Cumulative Brightness Function are presented (section 4). After, the classification algorithms are briefly illustrated (section 5). The experimental results obtained on real image sequences

acquired during football matches of the Italian Serie A are described in section 6. Finally, conclusions and future works are reported in section 7.

2 System Overview

The multi-camera environment consists of a real system installed in the "Friuli" stadium situated in Udine (Italy). This prototype permits to detect automatically "offside" during the football match [15]. The system is composed by six high resolution (Full HD) cameras (labeled as FG_i , where i indicates the i -th cameras) placed on the two sides of the pitch. This location assures double coverage of almost all the areas by either adjacent or opposite cameras. In figure 1 the location of the cameras is shown. The acquired images are transferred to six processing nodes by fiber optic cables. The acquisition process is guided by a central trigger generator that guarantees synchronized acquisition between all the cameras. Each node, using two hyper-threading processor, records all the images of the match on its internal storage unit, displays the acquired images and, simultaneously, processes them with parallel threads, in an asynchronous way with respect to the other nodes. The six processing nodes, are connected to a central node, which has having the supervisor function. It synchronizes the data coming from nodes and performing high level processing. The figure 2 shows the six images acquired from the six nodes linked to the cameras located around the pitch (see figure 1). Each nodes uses a motion segmentation algorithm [8] based on statistical background subtraction. Information relative to moving objects are the used to perform human blob detection. The player blobs represent the starting point of the classification step. We have evaluated the performance of different combination of unsupervised classifier and color feature applied in a multi-camera environment.

3 Feature Selection

In order to separate players in different classes, they should be represented by a features vector able to emphasize both intra-class analogies, and inter-class differences. Moreover, the selected features should be as well scale invariant (images of players could have different size according to the geometry of acquisition sensors and their position in the field), rotation invariant (usually players are standing, but sometimes they can appear slanted on the field), and also quickly extractable (real time processing is often a fundamental requisite for sport analysis applications). Starting from these requirements, we have tested three different feature sets that satisfy the above mentioned conditions:

RGB histograms: the RGB histogram is a combination of three 1-dimensional histograms based on the R,G and B channels of RGB color space.

rg histograms: in the normalized histograms the chromaticity components r and g describe the color information in the image; it is robust to light variations in luminance;

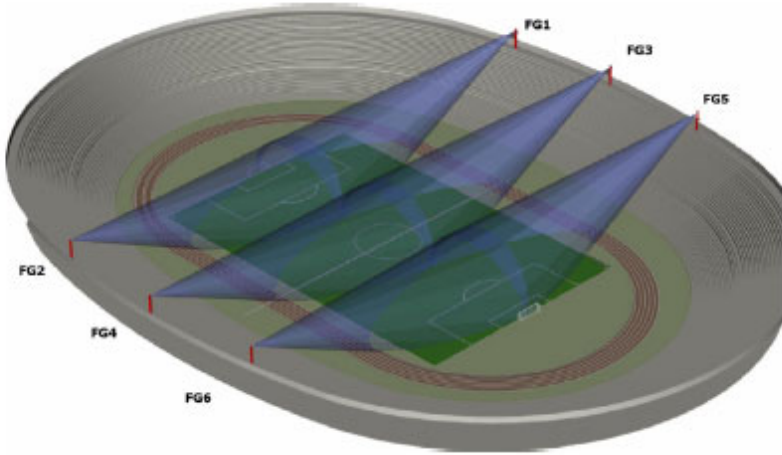


Fig. 1. Cameras' location around the pitch



Fig. 2. Snapshot of the six camera views after processing step

Transformed RGB histograms: each channel is normalized independently, obtaining for each channel a distribution where $\mu = 0$ and $\sigma = 1$.

Note that the last two sets introduce partial independence from light conditions. However, this is not sufficient to overcome problems related to the use of different sources: in presence of images coming from different cameras, usually the color appearance of the same actor changes radically from one image to another one. This happens also in presence of differently lighted regions in the same images (players of the same team positioned in shadowed/sunny regions). For this reason in the next section [4](#) we propose an approach to uniform color appearance in different images coming from uncalibrated cameras.

4 Cumulative Brightness Transfer Function

The main aim of this procedure is to relate all color distributions to a refer one, previously selected. So, after this step, players of the same class should have the same appearance, independently from the acquisition camera settings, and from the color content of the image. We evaluated the histograms in the RGB channels for all the segmented images of each of the N cameras. The histograms were generated by using 64 bins for each channel. We wanted to estimate BTFs between the reference camera and the others $N - 1$. We propose the generic algorithm relative to two different cameras and FOVs and we use it between each cameras and the reference one. For each couple of images from different FOVs (i_1, j_2) we want to estimate a BTF $f_{1,2}$ such that, for each couple of images (i_1, j_2) , given the brightness values $B^{i_1}(k)$ and $B^{j_2}(k)$ we have $B^{j_2}(k) = f_{1,2}(B^{i_1}(k))$ where $k = 0, \dots, 63$ represents the number of bins, $i_1 = 1, \dots, M$ represents the number of images in the camera, $j_2 = 1, \dots, N$ the number of images in the reference camera. For each possible couple of histograms (i_1, j_2) we evaluated the brightness transfer function

$$f_{i_1 j_2}(B^{i_1}(k)) = B^{j_2}(k) \quad (1)$$

using the inverted cumulative histogram, that is

$$f_{i_1 j_2}(B^{i_1}(k)) = H_{j_2}^{-1}(H_{i_1}(B^{i_1}(k))) \quad (2)$$

Using this concept we evaluate the cumulative BTF (CBTF) proposed in [7]. The generation of the CBTF involves an amalgamation of the training set before computing any BTFs. An accumulation of the brightness values is computed on all the training images of the generic camera obtaining a cumulative histogram \widehat{H}_1 . The same is done for all the corresponding training images of the reference camera obtaining \widehat{H}_2 . The CBTF $\widehat{f}_{1,2}$ is

$$\widehat{f}_{1,2}(B^1(k)) = \widehat{H}_2^{-1}(\widehat{H}_1(B^1(k))) \quad (3)$$

also in this case evaluated by using the inverted cumulative histogram. Notice that the same algorithm could be implemented starting from different part of the same FOV in order to smooth different color appearance due to different illuminations (play field with shadow and non uniform brightness).

5 Classification Algorithms

In our experiments we have implemented and tested three methodologies, belonging to different categories, to perform an unsupervised classification of players in five different classes (two teams, two goalkeepers, and officials): MBSAS (sequential algorithm), BCLS (competitive algorithm) and K-means (hard-clustering algorithm). We remain the reader to [9] for a detailed explanation of them. The algorithms need the definition of a proximity measure $d(x, C)$, a threshold of similarity th and the maximum number of clusters q . Euclidean distance has been used for similarity evaluations, while the maximum number of cluster has



Fig. 3. Example of 4 different cameras FOV with variable illumination conditions

been fixed to five according to our domain constraint, as previously remarked. The values assumed for the thresholds and the other specific parameters will be explicitly mentioned in the experiments section.

MBSAS: it is a sequential algorithm; vectors are presented twice, the first time for the representatives creation, and the second one for the assignment of all vectors to classes; it is dependant from the presentation order; each cluster is represented by a vector called prototype that can be updated at each presentation in the test phase.

BCLS: it is a competitive algorithm; representatives are randomly initialized; vectors are presented twice, the first time for the representatives updating (only the winner representative is updating at each presentation), and the second one for the assignment of all vectors to classes; it is dependant from the presentation order, and from the initial position of representatives. Again, each prototype can be updated at each presentation in the test phase.

Isodata (or k-means): it is a hard-clustering algorithm; representatives are randomly initialized; vectors are presented continuously until the representatives remain unchanged; at each presentation representatives are updated in function of the difference with the presented vector; it is dependant from the initial position of representatives. Representatives can be continuously updated in the test phase.

In the following section [6](#), we present results obtained by crossing the different feature sets with the above mentioned unsupervised algorithms.

6 Experimental Results

We have tested the proposed algorithms with different sequences acquired during real football matches of the Italian serie A championship, acquired in different

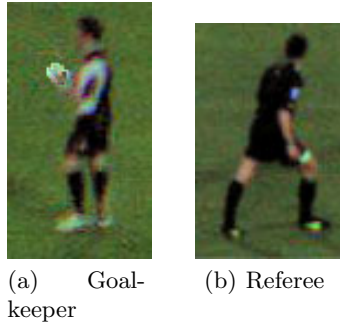


Fig. 4. Example of hard-distinguish players

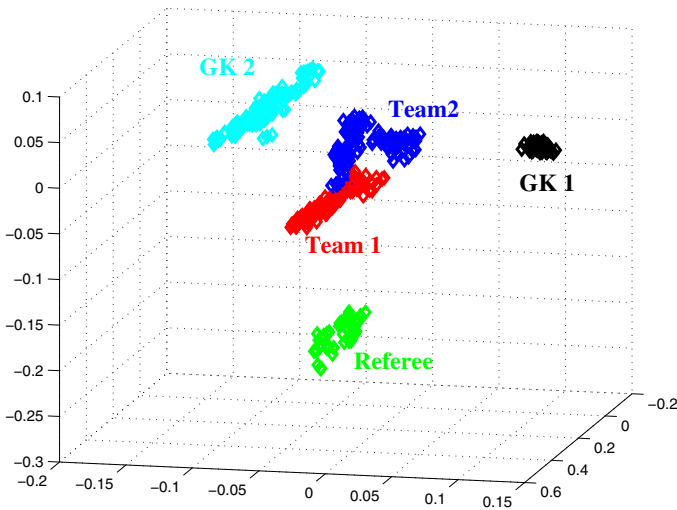


Fig. 5. PCA decomposition of Training Data

light conditions. The training set has been created by collecting a great number of player feature vectors, randomly selected (many times repeated) from real football images, with the care of including players positioned in different parts of the play field (to ensure the inclusion of goalkeepers and linemen referees). This feature set has been used during the training phase of the classifiers; each cluster has been represented by means of a feature vector ('representative' of the cluster). Then, at runtime each segmented player is provided to the classifier for the test phase. However, in this kind of applications, each game is a different case, and overall results could be misleading. For example, in a match we can have well-contrasted uniforms, with well separated classes, while in another one the classes could overlap in the feature space. For this reason in the following we present results obtained both on several matches (for testing the training phase) and on a single, random selected, match (for the test phase evaluation). Before

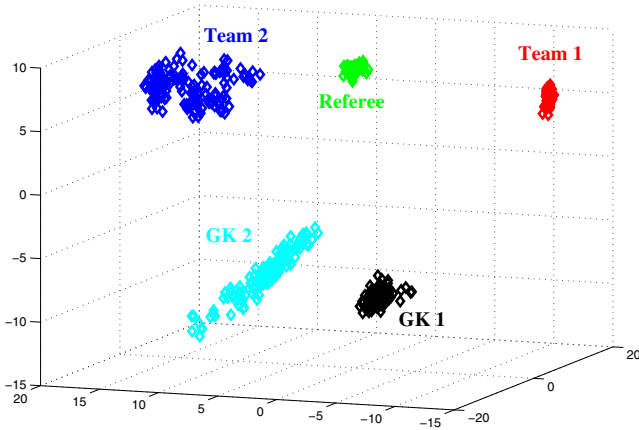


Fig. 6. PCA decomposition of Training Data after CBTF

Table 1. Reliability of the training procedure

		RGB	rg	T-RGB
No	MBSAS	71.24%	86.22%	93.12%
	BCLS	77.77%	87.33%	94.78%
CBTF	K-Means	81.43%	88.04%	95.31%
	Overall	78.99%	87.31%	94.96%
CBTF	MBSAS	74.12%	87.56%	94.31%
	BCLS	79.38%	89.45%	95.11%
	K-Means	85.83%	91.33%	97.17%
	Overall	82.13%	89.96%	96.18%

analyzing the results, here we report the processing parameters values for each algorithm (64-bin histograms have been used).

- **MBSAS**: th=0.5
- **BCLS**: $\mu = 0.2$, epochs=10000, exit th=0.01
- **K-means**: k=5, exit th=0.01

In the first experiment we have compared the capability of the training procedure to correctly detect the output clusters according to the different feature sets. For this purpose we carried out 10 experiments on 10 different matches; for each of them, about 1800 actors (players, goalkeepers and referees) images have been randomly collected in the training set, and provided to the algorithms. Note that these images have been acquired by different cameras, so there could be some differences in light conditions, as well as in color appearance. An example image, with four FOVs in which the illumination is variable, could be seen in fig. 3. Before to start with color feature extraction as explained in section 3 we have evaluated the different classes configuration (Team one, Team two, Goalkeeper one, Goalkeeper two and referee) in the original data and in the

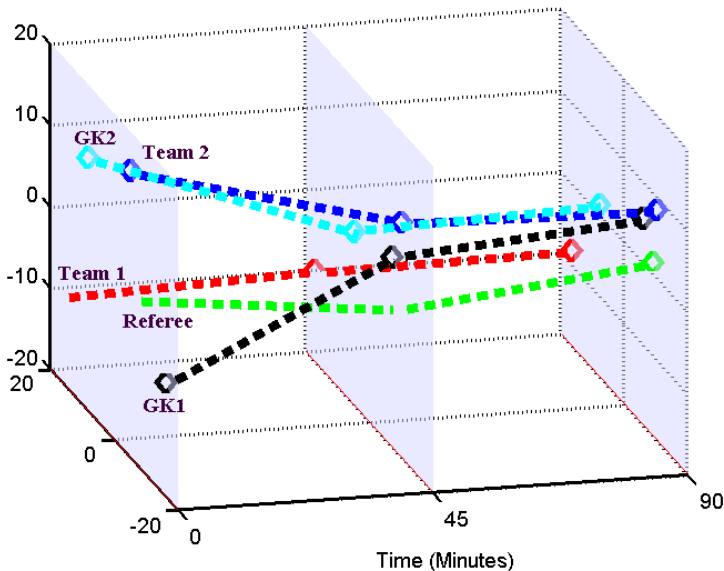


Fig. 7. Classes Temporal Evolution over a whole match

Table 2. Overall performance of the classifiers with Transformed RGB histograms

	MBSAS	BCLS	K-Means
NO CBTF	72.21%	81.33%	84.65%
CBTF	76.19%	84.24%	87.38%

transformed one (by means of CBTF). Figure 5 shows the original training data configuration by PCA decomposition. In figure 6 the transformed data are plotted using PCA decomposition technique. As the reader can see the CBTF (transformed data in figure 6) permits to separate the classes without overlapping and the relative cluster are better spaced.

Overall results of training both in presence/absence of preprocessing based on CBTF are presented in table 1. As it can be noted, the best overall results have been reported by using the Transformed RGB features in presence of CBTF preprocessing. It is probably due to the spectral invariancy introduced by them, while more sensible features, like simple RGB histograms, perform worse. However, the perfect separation of clusters has not been obtained for all sequences: by accurately observing images, in some football matches we noted that some clusters are really difficult to distinguish even for humans. For example, sometimes one of the goalkeepers was dressed in a very similar way with the referee, while in another match a goalkeeper was dressed like players of opposite team. In this case a correct classification based only on spectral information (without considering the player position in the play field) is really difficult also for human. In fig. 4 an example of two ambiguous classes is illustrated. Unfortunately, from

Table 3. Temporal analysis of performance of the classifiers

	MBSAS	BCLS	K-Means
0' - 15'	76.77%	85.34%	89.21%
16' - 30'	77.73%	84.14%	87.19%
31' - 45'	76.11%	84.72%	86.96%
46' - 60'	73.96%	81.62%	86.41%
61' - 75'	74.85%	80.81%	84.55%
76' - 90'	72.17%	79.92%	83.18%

the experience collected in our experiments in the last years, after viewing several games, we can assert that this situation (referees and goalkeepers dressed in similar way) is almost common in football games, and it drives our efforts into the direction of introducing a check of the player relative positions to make the classification more robust.

Starting from the results of these experiments, that demonstrates the better performance carried out by using the Transformed RGB histograms, we concentrate our efforts in order to detect the best unsupervised classifier (using transformed RGB as features set). In the second experiment we compared the three unsupervised classifiers during the test phase, i.e. we evaluated their capability to properly classify each actor according to the previously detected classes. In table 2 the overall performance obtained in the test phase are presented. Again, experiments have been performed both in absence/presence of CBTF preprocessing. We can note that K-means based approach seems to outperform the other ones, with a classification rate over then 87%.

In table 3 the results of overall classification as a function of the time are shown. These results coming from a new experiment: for a single match a ground truth was created by considering patches of players at a fixed time instants. In particular we considered 1800 patches (from six cameras after CBTF transform) extracted every 15 minutes, for a total of $1800 \cdot 6 = 10800$ patches. As evident all the classification performances are more reliable in the first minutes and then they decrease (not strongly) along the time. The temporal variation of cluster configuration has to be expected during the football match, in particular in outdoor contexts. The great duration of the event (90 minutes plus interval) is accomplished by variation in light conditions. An observation about our experiments needs to be remarked: we trained the classifier *before* the kick off, during the pre-match operations. This training remain unchanged for all the match. Probably the effects of class configuration changes could be mitigated if the training was carried out again at the beginning of second half. However, it is not the best practical solution, it could be unpracticable in real time applications; moreover the variation could be sudden (switch on/off of artificial lights), in an arbitrary instant, so it cannot be forecasted. In figure 7 is plotted the class configuration at the begin of the match, immediately after the half time interval, and at the end of the match. As evident, some classes greatly changed, while others changed in a less evident way. However at the end of the match the clusters are closer and this confirms the results obtained in table 3.

7 Conclusion and Future Work

In this paper, different color descriptors, an innovative technique for smoothing color difference between different FOVs and unsupervised classifiers are studied in the football multi-views environment. We evaluated three different color descriptors (RGB, normalized RGB and Transformed RGB histograms) we transformed them by a CBTF in order to mitigate the FOVs difference, and three unsupervised classifiers (MBSAS, BCLS and k-means). Other descriptors, as Color Sift and Moments, have not been considered since they are not reliable in presence of highly deformable objects, such as moving players. After the experiments on real videos, we can conclude that the better performance were carried out using the Transformed RGB histograms combined with k-means classifier after the application of CBTF on the original data. As a future work, the analysis of unsupervised team discrimination here proposed could be further improved by considering different feature sets, and different classifiers. One weak point of our experiments was that similar uniforms can be seldom found and all the methods suffer in separating different classes. This results was expected since the considered color descriptors are based on histogram evaluations that lose the spatial information on the color distribution. The next step will be to investigate on color features that can be applied to highly dynamic moving objects, not subject to rigid motion constraints, such as connected graphs of color histograms or weighted histograms on segmented body parts.

References

1. Assfalg, J., Bestini, M., Colombo, C., Del Bimbo, A., Nunziati, W.: Semantic annotation of soccer videos: automatic highlights identification. *CVIU* 92, 285–305 (2003)
2. Beetz, M., Bandouch, J., Gedikli, S.: Camera-based observation of football games for analyzing multi-agent activities. In: *AAMAS*, pp. 42–49 (2006)
3. Ekin, A., Tekalp, A., Mehrotra, R.: Automatic soccer video analysis and summarization. *IEEE Trans. on Image Processing* 12, 796–807 (2003)
4. Hayet, J., Mathes, T., Czyz, J., Piater, J., Verly, J., Macq, B.: A modular multi-camera framework for team sports tracking. In: *AVSS*, pp. 493–498 (2005)
5. Mazzeo, P.L., Spagnolo, P., d’Orazio, T.: Object tracking by non-overlapping distributed camera network. In: Blanc-Talon, J., Philips, W., Popescu, D., Scheunders, P. (eds.) *ACIVS 2009*. LNCS, vol. 5807, pp. 516–527. Springer, Heidelberg (2009)
6. Naemura, N., Fukuda, A., Mizutani, Y., Izumi, Y., Tanaka, Y., Enami, K.: Morphological segmentation of sport scenes using color information. *IEEE Tr. on Br.* 46, 181–188 (2003)
7. Prosser, B., Gong, S., Xiang, T.: Multi camera matching using bi-directional cumulative brightness transfer functions. In: *BMVC 2008* (2008)
8. Spagnolo, P., D’Orazio, T., Leo, M., Distanti, A.: Moving object segmentation by background subtraction and temporal analysis. *Im. Vis. Comp.* 24(5), 411–423 (2006)
9. Theodoridis, S., Koutroumbas, K.: *Pattern Recognition*. Academic Press, London ISBN 0-12-686140-4

10. Vandembroucke, N., Macaire, L., Postaire, J.: Color image segmentation by pixel classification in an adapted hybrid color space.application to soccer image analysis. *CVIU* 90, 190–216 (2003)
11. Xu, M., Orwell, J., Lowery, L., Thirde, D.: Architecture and algorithms for tracking football players with multiple cameras. *IEE Proc. Vis. Im. and Sign.* 152, 232–241 (2005)
12. Xu, Z., Shi, P.: Segmentation of players and team discrimination in soccer videos. *IEEE int. Work. VLSI Design Video Tech.*, 121–212 (2005)
13. Yu, X., Leong, H., Lim, J., Tian, Q., Jiang, Z.: Team possession analysis for broadcast soccer video based on ball trajectory. In: *ICICS-PCM*, pp. 1811–1815 (2003)
14. Zhang, D., Shih-Fu, C.: Real-time view recognition and event detection for sports video. *J. Vis. Commun. Image r.* 15, 330–347 (2004)
15. D’Orazio, T., Leo, M., Spagnolo, P., Mazzeo, P.L., Mosca, N., Nitti, M., Distante, A.: An Investigation Into the Feasibility of Real-Time Soccer Offside Detection From a Multiple Camera System. *J. Cir. Sys. Video* 19(12), 1804–1818 (2009)

SUNAR

Surveillance Network Augmented by Retrieval

Petr Chmelar, Ales Lanik, and Jozef Mlich

Brno University of Technology, Faculty of Information Technology,
Bozotechnova 2, 612 66 Brno, Czech Republic
{chmelarp, ilanik, imlich}@fit.vutbr.cz
<http://www.fit.vutbr.cz>

Abstract. The paper deals with Surveillance Network Augmented by Retrieval (SUNAR) system – an information retrieval based wide area (video) surveillance system being developed as a free software at FIT, Brno University of Technology. It contains both standard and experimental techniques evaluated by NIST at the AVSS 2009 Multi-Camera Tracking Challenge and SUNAR performed comparably well.

In brief, SUNAR is composed of three basic modules – video processing, retrieval and the monitoring interface. Computer Vision Modules are based on the OpenCV Library for object tracking extended by feature extraction and network communication capability similar to MPEG-7. Information about objects and the area under surveillance is cleaned, integrated, indexed and stored in Video Retrieval Modules. They are based on the PostgreSQL database extended to be capable of similarity and spatio-temporal information retrieval, which is necessary for both non-overlapping surveillance camera system as well as information analysis and mining in a global context.

Keywords: SUNAR, wide area, surveillance, video analytics, retrieval, similarity, tracking, trajectory, integration.

1 Introduction

Nowadays, there is a lot of data produced by wide area surveillance networks. This data is a potential source of useful information both for on-line monitoring and crime scene investigation. Machine vision techniques have dramatically increased in quantity and quality over the past decade. However, the state of the art still doesn't provide the satisfactory knowledge, except some simple problems such as people counting and left luggage or litter detection.

Justin Davenport in Evening Standard [6] showed statistics of crime-fighting CCTV cameras in Great Britain. The country's more than 4.2 million CCTV cameras caught (in 2007) each British resident as many as 300 times each day. BBC News [1] informed that half a million pounds a year was spent on talking cameras helping to pick up litter. Yet 80% of crime is unsolved. Well, we agree that high quality crime investigation is the best prevention.



Fig. 1. An example of a successful camera pair handover

The idea was to create an automated system for object visual detection, tracking and indexing that can reduce the burden of continuous concentration on monitoring and increase the effectiveness of information reuse by a security, police, emergency and firemen (or military) and to be useful in the accident investigation. The task is to perform the analysis of the video produced by a camera system with non-overlapping field of views. The analysis, based on cleaned, integrated, indexed and stored metadata, is of two types – on-line used for identity preservation in a wide area; and off-line to query the metadata of the camera records when an accident, crime, a natural or human disaster (war) occurs.

In 2006, we have started to develop an IR-based multi-camera tracking system to be at the top of the state of the art. We have taken part in several projects (CARETAKER [4]) and evaluations (TRECVID [19]) concerning similar problems. However, the AVSS 2009 Multi-Camera Tracking Challenge [20] was the first evaluation campaign that used the annotated Multiple-camera Tracking (MCT) Dataset from the Imagery Library for Intelligent Detection Systems (i-LIDS) provided by Home Office Scientific Development Branch (HOSDB) of the UK [16]. We have used the MCT video data and annotations to train and evaluate the SUNAR performance and it performed comparably well.

The paper is organized as follows. The introduction presents our motivation and ideas. An overview and design of the SUNAR system is described in the following section. Computer vision methods are described in section 3. Object identification, search and analysis techniques are described in section 4. The NIST performance evaluation of the SUNAR system is in section 5. State of the art is situated at the beginning of each section. The paper is concluded in section 6.

2 System Design

Although there are many multi-camera surveillance systems [10,7,12,13], we believe our approach outperforms the others, because those described in literature were not evaluated successfully [10,12], while those in praxis make many simplifying presumptions (e.g. traffic monitoring). Moreover, there is no need for a central or primary module [7] or some special hardware such as camera sensors [13]. Moreover, it is able to derive various useful information concerning the entire area under surveillance.

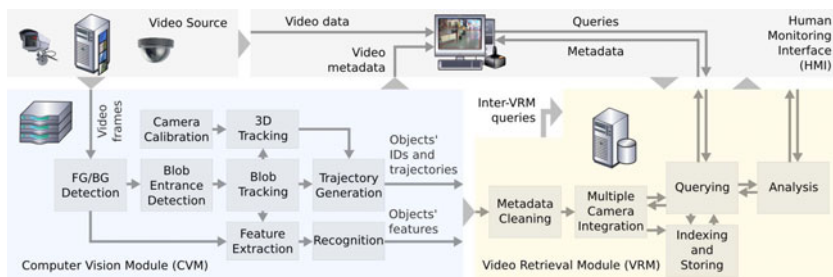


Fig. 2. Illustration of the multiple camera tracking process including the manual annotations

From the schematic perspective, SUNAR consists of the following modules, as illustrated in figure 2:

0. Source of video (any provider)
1. Computer Vision Modules (CVM)
2. Video Retrieval Modules (VRM)
3. Human Monitoring Interfaces (HMI)

The video source might be e.g. a camera or a video server and it is not a generic part of the system. Each module except the Human Monitoring Interface is responsible for capturing, analysis and retrieval in an appropriate part of the wide area under surveillance. Modules communicate basically only with their neighborhoods using the IP protocol. In this way, we can build a considerably large system, because no special central unit is necessary.

The input of the Computer Vision Module (CVM) is a video stream. We use OpenCV [8] for tracking and 3D calibration especially (if feasible). We have extended the OpenCV Blobtrack to be capable of feature extraction, object (and event) recognition and IP based video stream capability. The output of the CVM module is metadata of objects and the environment. It includes local identification of objects, its spatio-temporal location and changes (speed) and a description of objects – dimensions, shape, color, texture or other special features (e.g. license plate and face descriptor) similarly to MPEG-7 [9]. The description is complemented with a recognition of basic object classes (e.g. cars, trolleys, people and groups) and events (e.g. opposing flow and left luggage).

The main contribution of the proposed wide area system is in the Video Retrieval Module (VRM). The input of the module is metadata produced by CVMs. This metadata is cleaned and normalized in time and space (lighting, color bias and 3D parameters) and stored in the PostgreSQL database (www.postgresql.org). The primary function of the VRM is to identify objects – to integrate identifiers (ID) of objects in the wide area, based on the previous occurrence of an object and its appearance. This is accomplished by the use of information retrieval and video search methods based on metadata produced by CVMs as further described in section 4.

The Human Monitoring Interface is then capable not only of a simple monitoring the area, but also querying monitored objects based on their previous occurrences, visual properties and behavior. The behavior is derived from an object's trajectory, its interactions with the environment and mutual interactions based on statistical and data mining methods. This is illustrated in figure 1b.

3 Computer Vision Techniques

There are two major spheres we would like to evaluate – computer vision and surveillance information retrieval. The computer vision part is further divided in the object tracking, feature extraction and 3D calibration as illustrated in figure 2.

The computer vision is a broad but still underdeveloped area summarized by Sonka, Hlavac and Boyle in [14]. We concern on visual surveillance methods, especially on distributed surveillance systems, reviewed by Valera and Velastin [15] and CARETEKER deliverables [4].

The 3D camera calibration [14] is an optional technique in the IR based approach, when an exact 3D calibration is required, we use CARETAKER's KalibroU – a camera calibration program, based on Tsai's method [4]. Thus we concentrate more on tracking, feature extraction and object recognition.

3.1 Object Tracking

Object tracking [14] is a complex problem and it is hard to make it working well, in real (crowded) scenes as illustrated in figure 3. Discussed approach is based mainly on proved methods of object tracking implemented in the Open Computer Vision Library [8]. The tracking process is illustrated in figure 2. Background is modeled using Gaussian Mixture Models [8] as an average value of color in each pixel of video and the foreground is a value different to the background. We have been inspired by the approach developed by Carmona et al. [3].

Foreground is derived from background, which is modeled using Gaussian Mixture Models [8] as an average value of color in each pixel of video and the foreground is a value different to the background based on segmentation of the color in RGB color space into background, foreground and noise (reflection, shadow, ghost and fluctuation) using a color difference Angle-Mod cone with vertex located in the beginning of the RGB coordinate system. In this way, the illumination can be separated from the color more easily.

The other two modules – blob entrance and tracking are standard OpenCV Blobtrack functions [8]. The blob entrance detection tracks connected components of the foreground mask. The Blob tracking algorithm is based again on connected components tracking and Particle filtering based on Means-shift resolver for collisions. There is also a trajectory refinement using Kalman filter as described in section 4.

The trajectory generation module has been completely rewritten to add the feature extraction and TCP/IP network communication capability. The protocol is based on XML similarly to MPEG-7 [9]. The objects' ID and trajectory is in this way delivered to a defined IP address and service (port 903).

3.2 Feature Extraction and Object Recognition

There are more possibilities how to make a multi-camera surveillance system [7][12][13]. Because of our goal – to acquaint as much information about objects as possible, we use visual surveillance information retrieval instead of (multi-)camera homography or handover regions as in [7]. Moreover, the area might be large and objects will occlude in those regions.

Although there are many types of features to be extracted [14], primarily we use descriptors based on the visual part of MPEG-7 [9]. We try to avoid color descriptors only, as in [13], because most of airport passengers (at least on British Isles) wear black coats and there is a lot of dark metallic cars there.

However, we have adopted color layout concept, where each object is resampled into 8x8 pixels in Y’CbCr color model. Then, the descriptor coefficients are extracted zig-zag from its Discrete cosine transform similarly to JPEG. Other (texture) descriptor is based on extraction of energy from (Fourier) frequency domain bands defined by a bank of Gabor filters [9].

For the object classification we use also local features (such as SIFT and SURF) and a simple region (blob) shape descriptor. The shape together with previously described object metadata then acts as an input of a classification algorithm in the recognition procedure of the CVM. The object recognition process is based on 2 popular machine learning methods – AdaBoost and Support vector machines (SVM), the OpenCV [8] implementation. The system has a simple training GUI to mark an object by a simple click while holding a key to associate a blob to its appropriate class or to change the class of a misclassified sample.

To avoid this, CVM may use AdaBoost object detection based on Haar features, similarly to the OpenCV face detection. Unfortunately, there are just a few faces to be detected in the standard TV resolution video and camera setup similar to the MCT dataset. The detector is followed by MPEG-7 Face recognition descriptor [9]. Other face recognition approaches will be compared in the future to allow a more precise and consistent object tracking and recognition in low-resolution images and video. Thus, we concentrate more on retrieval methods at the moment.

4 Surveillance Information Retrieval

Although there were published basics of wide area surveillance systems with non-overlapping fields of view [10], these systems suffer from multiple deficiencies caused by the curse of dimensionality – e.g. they allow only simple handover regions [7] or they are unable to act in a crime investigation process [12][13], because the real recordings are too massive and of low quality to be analyzed efficiently (as in CSI NY series).

The metadata coming from CVMs – local IDs, trajectories and object description must be cleaned, integrated, indexed and stored to be able of querying and analyzing it, as illustrated in figure [2].

4.1 Metadata Cleaning

The preprocessed data is supposed to be incomplete or duplicate, biased and noisy. Thus, moving objects are modeled as dynamic systems in which the Kalman filter optimally minimizes the mean of error [5] and it can fill in the missing information (position and velocity) for a few seconds in case the object has been occluded, for instance [4].

At the cleaning step, SUNAR stores metadata representing moving objects and information about the environment under surveillance.

4.2 Indexing and Storing

The database model consists of three database schemes in the SUNAR database – Process, Training and Evaluation according to their purpose. All schemes contain three main tables that correspond to the fundamental concepts – Object, Track and State (as in our former work [5]). Object is an abstract representation of a real object (having a globally unique ID), it is represented by its states. A state consists of two types of features – visual properties (as described in section 3) and spatio-temporal features. The latter are represented by location and velocity of an object at a moment. A track is a sequence of such states in a spatio-temporal subspace of the area under surveillance followed by one camera.

The training scheme contains also tables containing statistics and classification models according to the method used. For instance, a simplified Bayesian model table contains columns for source and destination camera IDs, in which objects are passing through. Next columns represent the number of training samples, a prior probability, averages and variances of handover time, trajectory states and visual features. Trajectories are summarized as a weighted average of cleaned states, where the weight is highest at the end of the trajectory. If cameras are overlapping, the handover time may be negative. The average and variance of different feature descriptors acts as the visual bias removal (illumination, color, viewpoint and blob size calibration) for the integration step.

4.3 Multiple Camera Integration

The training schema described before is rather simplified. In fact, we use Gaussian Mixture Model and Support Vector Machine [14,8] models of the (inverted) Kalman filter state as described in our previous work [5]. The inverted state is computed using Kalman filter in the opposite direction the object moved through one camera subspace followed by one camera. The goal of this trick is the classification of the previous subspace (camera) in which it was seen last time most probably.

The object identification then maximizes the (prior) probability of a previous location (camera) multiplied by the normalized similarity (feature distance without bias) to previously identified objects according to average time constraints and visual features in the database [5,10]. More formally an optimal

¹ Available at www.fit.vutbr.cz/research/view_product.php.en?id=53

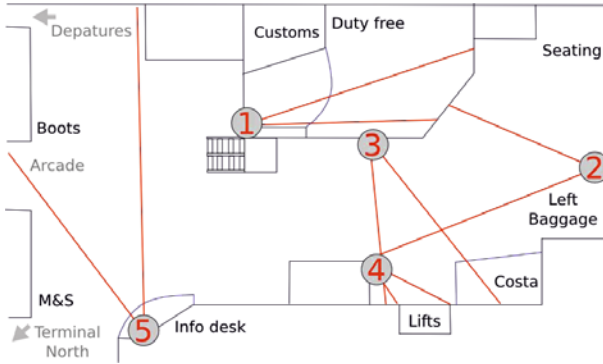


Fig. 3. i-LIDS multiple camera tracking scenario definition provided by HOSDB

identifier (k^*) of the object in the wide area is based on its previous occurrence (spatio-temporal, o) and its state (appearance, s):

$$k * (o, s) = \operatorname{argmax}_k P(k|o, s) \approx P(o|k)P(s|k) \quad (1)$$

Because of this, we must (approximately) know the camera topology. The figure 3 is suitable enough for the learning step. We have used annotations provided by the HOSDB on i-LIDs MCT dataset. There are 5 cameras and several areas from where a new object can enter.

The object appearance and bias is automatically learned (or summarized) using Gaussian mixture models [8] or optionally SVM. The probability $P(s|k)$ is then determined by a similarity search (the distance is normalized using the sigmoid) with respect to the expected bias, which is simply subtracted.

4.4 Querying

The SUNAR queries are of two types – on-line used for instantaneous condition change and especially for identity preservation as described above; and off-line queries, able to retrieve all the metadata from processed camera records in the wide area after an accident, crime or a disaster happens.

We can distinguish two types of operations: environmental and trajectory operations. Environmental operations are relationships of an object's trajectory and a specified spatial or spatio-temporal environment, such as enter, leave, cross, stay and bypass [2,5]. Trajectory operations look for relationships of two or more trajectories restricted by given spatio-temporal constraints, such as together, merge, split and visit.

We have also implemented [2] similarity queries based on MPEG-7 features in the PostgreSQL database as a vector (array) distance functions – Eukleidean (Mahalanobis), Chebyshev and Cosine distance.

² Available at www.fit.vutbr.cz/research/view_product.php.en?id=73



Fig. 4. Illustration of the multiple camera tracking process of the SUNAR system including manual ground truth annotations provided by HOSDB and NIST

4.5 Analysis

We perform several types of video analysis, mainly classification and clustering as illustrated in figure 1b. The first type is the modeling based on visual appearance of an object (color layout, blob) using Gaussian Mixture Models (GMM, [8]).

Second, we perform trajectory classification based on Gaussian Mixture Models as needed for the multiple camera identification as in section 4 and Hidden Markov Models (HMM). In the article [11] we selected few scenes, where some easily recognizable human behavior occurs. For example, one concept represents if people go through turn pikes or not. The HMM are trained on such classes. The trajectory which doesn't fit any HMM model (with respect to some threshold) is considered to be abnormal. In addition, SUNAR uses velocity and acceleration as training features, which describe and discover some abnormalities better (jump over).

Moreover, using the spatio-temporal queries, we can discover splitting and merging objects, opposing flow (together with GMM and aggregate functions) or an object put (operations enter, split, leave and stay).

5 Evaluation

The previous evaluations such as Performance Evaluation of Tracking and Surveillance (PETS [17]) dealt with other aspects of computer vision than multiple camera surveillance with non-overlapping camera fields of view. They either dealt with classical single camera tracking or they have concerned more on the event detection as Classification of Events, Activities, and Relations. For instance, events so-called left baggage, split, hug, pointing, elevator no entry are detected in the TRECvid Surveillance Event Detection evaluation [19].

The AVSS 2009 Multi-Camera Tracking Challenge [20] was the first evaluation campaign that used the annotated Multiple-camera Tracking (MCT) Dataset

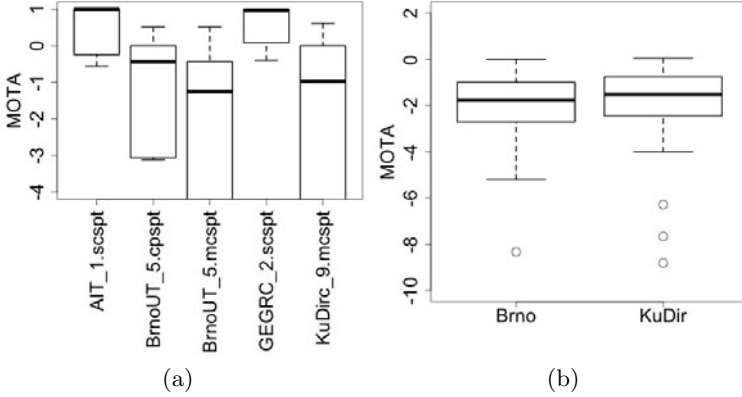


Fig. 5. The NIST’s single camera (a) and multiple camera (b) single person tracking MOTA evaluation medians

from the Imagery Library for Intelligent Detection Systems (i-LIDS) provided by Home Office Scientific Development Branch (HOSDB) in the UK [16]. We have used the MCT video data and annotations to train and evaluate the SUNAR performance. The data set consists of about 44 hours of video recorded by five cameras at the London Gatwick Airport.

The task is defined as: Given 5 in situ video frames with bounding box data specifying a person to be tracked, track the person in 5, 2 or 1 camera views by outputting bounding boxes [20].

We have participated in the compulsory Multi-Camera Single Person Tracking (MCSPT) and Camera Pair Single Person Tracking (CPSPT). The illustration of the data and the area under surveillance is in figures 2, 3 and 1. For more details see [20].

According to Johnatan Fiscus’s and Martial Michels’s presentation at the 2009 AVSS conference, [20] and received evaluated submissions, they used especially the Multiple Object Tracking Accuracy (MOTA, [18]) metric. The correct detection here is when it states:

$$MOTA = 1 - \frac{\sum_{t=1}^{N_{frames}} (c_m(m^{(t)}) + c_f(fp^{(t)}))}{\sum_{t=1}^{N_{frames}} N_G^{(t)}} \quad (2)$$

The $G_i^{(t)}$ is the ground truth bounding box of an object i at (frame or) time t , the $D_i^{(t)}$ is a (SUNAR) system detection accordingly. Else the detection is false positive $fp^{(t)}$, or missed $m^{(t)}$ if there is no system detection at time t . Then the MOTA is defined as [2]. Where c_m and c_f are weights (=1 this time) and N_G is the number of ground-truth objects at time t . The perfect MOTA is 1, but it may go down to $-\infty$ because of false alarms [20]. The (median) MOTA results for single camera and multiple cameras are illustrated in the figure 5. There the camera pair run (BrnoUT_5.cpspt) was better than our multiple camera run (BrnoUT_5.mcspt) because of the state space to be searched. Thus the single

Table 1. Multiple camera tracking results - MOTA

MOTA	Brno	KuDir
Test Set Average	-1.183	-1.400
Track Averaged Mean	-2.052	-2.072
Track Averaged Median	-1.770	-1.517

Table 2. The primary to Secondary Camera subject Re-Acquisition (SCRA) metric table

		Sec. RA - GT					Sec. RA - Brno					Sec. RA - KD				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Primary cam	1	9					1					0				
	2		8					2					0			
	3			7					0					0		
	4				1					0					0	
	5					9					0					0

camera (scspt) runs are incomparable to multiple camera runs. In table 1, only MCSPT results are depicted.

The table 2 also shows that using standard precision/recall metrics, our results are slightly better than other results [20]. Moreover, using the multiple camera (summarized binar) metric – the (primary to) Secondary Camera subject Re-Acquisition (SCRA, [20]) shows that SUNAR slightly outperformed the other systems in absolute numbers, which may be seen in table 2. The CPSPT task results were similar to the table above, but we have been the only participants there. The illustration of the task is in figure 1. In both figures 1 and 4 (an illustration of a MCSPT tracking trial), the bounding boxes and trajectories are of five colors. Blue means non-occluding reference (ground truth), yellow an occluding reference. The Green box and trajectory shows a correct detection, red represents a missed detection and the orange color is for false alarms.

6 Conclusions

This paper presents a state of the art SUNAR surveillance system based on visual information retrieval in theory and praxis (using free software). In contrast to other approaches, we try to collect and index as much information as we can acquaint and manage it efficiently to avoid a continuous human CCTV monitoring and analysis of massive and low quality recordings in case of an accident.

The FIT, Brno University of Technology has taken part in many projects and evaluations concerning the public safety and visual surveillance, however the AVSS 2009 Multi-Camera Tracking Challenge [20] was the first public evaluation campaign concerning object tracking in a wide area under surveillance containing both camera setups – overlapping and non-overlapping field of views.

Although we are convinced the system works really good under certain circumstances and it outperformed the others especially in the Secondary Camera

subject Re-Acquisition (SCRA) metric at the AVSS conference, there are some issues.

Especially those concerning computer vision techniques – object detection, tracking and recognition performance in low quality video. First, the quality of the embedded OpenCV methods should be extended by (many) parameters tuning. Second, the problem is to find more reliable visual features necessary for the object re-identification, because almost everybody wears black at the airport and objects are represented by a few pixels. We suppose moving to the high-definition video will result in more precise event recognition, occlusion handling and feature extraction for the automated wide area surveillance.

References

1. BBC 'Talking' CCTV scolds offenders. BBC News (April 4, 2007)
2. Brakatsoulas, S., Pfoser, D., Tryfona, N.: Modeling, Storing and Mining Moving Object Databases. In: IDEAS (2004)
3. Carmona, E.J., Martinez-Cantos, J., Mira, J.: A new video segmentation method of moving objects based on blob-level knowledge. *Pattern Recognition Letters* 29, 272–285 (2008)
4. CARETAKER Consortium. Caretaker Puts Knowledge to Good Use. *Mobility, The European Public Transport Magazine* 18(13) (2008)
5. Chmelar, P., Zendulka, J.: Visual Surveillance Metadata Management. Database and Expert Systems Applications. In: Wagner, R., Revell, N., Pernul, G. (eds.) DEXA 2007. LNCS, vol. 4653, pp. 79–84. Springer, Heidelberg (2007)
6. Davenport, J.: Tens of thousands of CCTV cameras, yet 80% of crime unsolved. *Evening Standard* (September 19, 2007)
7. Ellis, T., Black, J., Xu, M., Makris, D.: A Distributed Multi Camera Surveillance System. *Ambient Intelligence*, 107–138 (2005)
8. Bradski, G.R.: *Learning OpenCV*, p. 555. O'Reilly, Sebastopol (2008)
9. ISO/IEC JTC1/SC29/WG11. MPEG-7 Overview (2004)
10. Javed, O., Shah, M.: *Automated Visual Surveillance: Theory and Practice*, p. 110. Springer, Heidelberg (2008)
11. Mlich, J., Chmelar, P.: Trajectory classification based on Hidden Markov Models. In: *Proceedings of 18th Int. Conf. on Computer Graphics and Vision*, pp. 101–105 (2008)
12. Qu, W., Schonfeld, D., Mohamed, M.: Distributed Bayesian multiple-target tracking in crowded environments using multiple collaborative cameras. *EURASIP J. Appl. Signal Process* (1) (2007)
13. Qureshi, F.Z., Terzopoulos, D.: Multi-camera Control through Constraint Satisfaction for Persistent Surveillance. In: *IEEE Conf. on Advanced Video and Signal Based Surveillance*, pp. 211–218 (2008)
14. Sonka, M., Hlavac, V., Boyle, R.: *Image Processing, Analysis, and Machine Vision*, 3rd edn., p. 800. Thomson Engineering, Toronto (2007)
15. Valera, M., Velastin, S.A.: Intelligent distributed surveillance systems: a review. *Vision, Image and Signal Processing, IEE Proceedings* 152(2), 192–204 (2005)
16. HOSDB. Home Office Multiple Camera Tracking Scenario data, scienceandresearch.homeoffice.gov.uk/hosdb/cctv-imaging-technology/video-based-detection-systems/i-lids [cit. 2009-11-17]

17. PETS: Performance Evaluation of Tracking and Surveillance, www.cvg.rdg.ac.uk/PETS2009 [cit. 2009-11]
18. Kasturi, R., et al.: Framework for Performance Evaluation of Face, Text, and Vehicle Detection and Tracking in Video: Data, Metrics, and Protocol. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 31(2), 319–336 (2009)
19. TRECVID Event Detection, www-nlpir.nist.gov/projects/tv2009/tv2009.html#4.1 [cit. 2009-11-17]
20. Fiscus, J., Michel, M.: AVSS 2009 Multi-Camera Tracking Challenge, www.itl.nist.gov/iad/mig/tests/avss/2009/index.html [cit. 2009-11-17]

Object Tracking over Multiple Uncalibrated Cameras Using Visual, Spatial and Temporal Similarities

Daniel Wedge*, Adele F. Scott, Zhonghua Ma, and Jeroen Vendrig

Canon Information Systems Research Australia

jeroen.vendrig@cisra.canon.com.au

<http://www.cisra.com.au>

Abstract. Developing a practical multi-camera tracking solution for autonomous camera networks is a very challenging task, due to numerous constraints such as limited memory and processing power, heterogeneous visual characteristics of objects between camera views, and limited setup time and installation knowledge for camera calibration. In this paper, we propose a unified multi-camera tracking framework, which can run online in real-time and can handle both independent field of view and common field of view cases. No camera calibration, knowledge of the relative positions of cameras, or entry and exit locations of objects is required. The memory footprint of the framework is minimised by the introduction of reusing kernels. The heterogeneous visual characteristics of objects are addressed by a novel location-based kernel matching method. The proposed framework has been evaluated using real videos captured in multiple indoor settings. The framework achieves efficient memory usage without compromising tracking accuracy.

Keywords: distributed tracking, surveillance, real-time systems.

1 Introduction and Related Work

The large sizes of modern surveillance camera networks mean that it may not always be possible for a human being to monitor every video stream in real-time. This presents the need for autonomous camera networks that can extract video content for either human users or higher-level autonomous processing. It is desirable in industry applications for these networks to operate in a decentralised manner in order to minimise setup time, for improved robustness and scalability, and so that in the case of active camera networks each camera may act as an autonomous agent. The objective of multiple camera tracking, or, multi-camera tracking, is to determine correspondences between observations of real-world objects seen by multiple cameras after object detection and single-camera tracking have been performed. We consider the case where cameras are uncalibrated and there is no knowledge of the network topology. Multi-camera tracking methods

* The author is currently with the University of Western Australia.

are typically divided into two broad categories based on the field of view (FOV) of each camera: common FOV methods [1] [2] where cameras' FOVs largely overlap, and disjoint FOV methods [4] [5] [6] where a camera "hands-off" the tracking of an object from the FOV of one camera to another camera. Traditional tracking methods such as Kalman filters are not appropriate when the topology of the camera network is unknown and cameras are uncalibrated [4].

One of the classic problems in multi-camera tracking over either overlapping or disjoint FOVs is the entry/exit problem, i.e., given that an object has left a FOV at a particular location, which camera is most likely to see the object next, where within that camera's FOV, and when? One solution to this problem was presented by Javed et al. in [7]. Visual characteristics of objects were first used to determine corresponding objects in different FOVs. Entry and exit points in each camera's FOV were then determined using kernel density estimation. Finally, optimal correspondences entry and exit points were determined using a maximum a posteriori (MAP) approach based on a bipartite graph. Javed's method works well with independent FOV scenarios without any inter-camera calibration. However, it is restricted by the following:

1. a training phase must be available where correspondences between tracks are known;
2. the entire set of observations must be available so hence, the method cannot be deployed for real-time applications; and
3. the changes in visual characteristics of objects between camera views are assumed to happen in the same, generally predictable way.

In this paper, we present a unified framework to solve the multi-camera tracking problem in both independent FOV and common FOV cases. We assume that objects have been independently tracked in each camera in a multi-camera network, as in [7], and then aim to determine correspondences between these tracks in a decentralised way, that is, without a centralised server. As in [7], our approach requires no camera calibration, or knowledge of the relative positions of cameras and entry and exit locations of objects.

In contrast to [7], we remove each of the constraints listed earlier. We use a kernel-based tracking algorithm, which creates kernels over the entire FOV of each camera rather than only at entry and exit points. Our system effectively performs unsupervised, online learning of a correspondence model by continuous collection and updating of tracking statistics. This allows the proposed algorithm to be performed in real-time with no need for a dedicated and supervised training phase, thereby lifting constraints 1 and 2. To enable this collection of tracking statistics we introduce the concept of reusing kernels, and show that by using this technique the memory usage of the system is bounded. We then introduce a location-based kernel matching method to address abrupt changes in visual characteristics of objects (often due to changes in object pose or camera angle) based on the historical data available through reusing kernels, thereby lifting constraint 3. This enables us to develop a lightweight, decentralised, multi-camera tracking solution with limited communication between cameras ensuring that an on-camera implementation is possible without requiring a coordinating server.

The main contributions of this paper are the methods of reusing kernels and location-based kernel matching.

This paper is organised as follows. In section 2, we propose the Signature-based Tracking Across Cameras (STAC) multi-camera tracking algorithm. In section 3, we describe the experimental setup and evaluation methods used. In section 4, we present results. Finally, in section 5 we make conclusions.

2 The STAC Algorithm

The proposed STAC algorithm runs on each camera in a multi-camera network as part of a parallel tracking framework. Each camera considers itself to be the *local camera* and other cameras as *foreign cameras*. We assume that each camera captures video at the same frame rate and that frames have been synchronised. Time is measured in units of *frames*. We assume that object detection and single-camera tracking have already been performed and the results of the single-camera tracking are the input of the STAC algorithm. This setup is shown in Fig. 1. Specifically, we assume that for each tracked object in the local camera we know: the object's centre (x, y) in pixels; the height and width of the object's bounding box in pixels; the object's signature in the most recent frame; and a unique identifier (track ID) for the track of the object in the local camera. A distance metric on the signature must be defined. For details of the signature type and metric used in our implementation see section 3.2.

The process of the STAC algorithm comprises three main steps:

1. finding visual similarities between pairs of objects;
2. finding spatial and temporal similarities between current pairs of objects and historical pairs of objects; and
3. determining correspondences between pairs of tracks.

Step 1 includes the novel concept of reusing kernels and the method of location-based kernel matching. Sections 2.1, 2.2 and 2.3 describe the details of steps 1, 2 and 3 above, respectively.

2.1 Finding Visual Similarities between Pairs of Objects

The STAC algorithm first attempts to find relationships between tracked objects' locations in the local camera's field of view and locations in foreign cameras' fields of view, based on the information received from the single-camera tracker. Let *kernels* represent locations of objects in a field of view, and a *linked pair of kernels* represent the visual and temporal relationship between two kernels. For simplicity, details of this process are described in the following subsections from the point of view of one camera and for one tracked object.

¹ In our experiments, we lift the assumption by employing a framerate compensation algorithm that examines timestamps of captured frames.

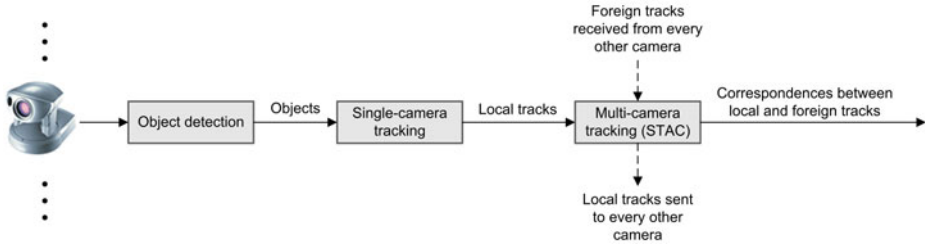


Fig. 1. A block diagram of the STAC algorithm running on a camera in a parallel multi-camera tracking framework

Constructing the local kernel and local track snapshot. We use a 2D Gaussian distribution to represent the kernel of a tracked object. This Gaussian form will be used for finding spatial and temporal similarities between linked pairs of kernels in section 2.2. The centre of the Gaussian distribution is located at the centre of the object, and the standard deviation in the x and y directions are equal to half the object’s height and width, respectively. Each local camera keeps a history of locally observed kernels of any tracked object previously observed in this camera.

For each frame in which the tracked object is visible, the *local track snapshot* of the tracked object is constructed. The local track snapshot contains: the kernel of the tracked object in the current frame of the local camera; the object’s signature in the current frame of the local camera; the local track ID, as determined by the single-camera tracker; and the current frame, f . This local track snapshot is sent to every camera in the network. Consequently, the local camera receives track snapshots from each foreign camera each frame, which we will call *foreign track snapshots*. The local camera stores the foreign track snapshots over time for use in linking kernels in the local camera to kernels in foreign cameras.

Reusing kernels. Creating a new kernel for each tracked object in every frame causes the number of kernels to grow rapidly, resulting in large demands on computing and memory resources. To overcome this, and to allow for historical tracking statistics to be collected, an existing kernel is reused if a tracked object has a similar position and size to a previously observed object. A similarity score, s , is calculated between the potential new kernel and each existing kernel, as detailed below.

The Euclidean distance, d , between the centres (x_1, y_1) and (x_2, y_2) of the potential new kernel and existing kernels, as well as the angle θ between the two centres, are determined by,

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \quad \theta = \tan^{-1} \left(\frac{y_2 - y_1}{x_2 - x_1} \right). \quad (1)$$

Then, the standard deviations σ_1 and σ_2 of the potential new kernel and existing kernel in the direction θ are calculated, which are given by (these equations are derived from the polar equation of an ellipse),

$$\sigma_1^2 = \frac{\sigma_{x_1}^2 \sigma_{y_1}^2}{\sigma_{y_1}^2 \cos^2 \theta + \sigma_{x_1}^2 \sin^2 \theta}, \quad \sigma_2^2 = \frac{\sigma_{x_2}^2 \sigma_{y_2}^2}{\sigma_{y_2}^2 \cos^2 \theta + \sigma_{x_2}^2 \sin^2 \theta}. \quad (2)$$

where σ_{x_1} and σ_{y_1} , are the standard deviations of the potential new kernel in the x and y directions respectively, and σ_{x_2} and σ_{y_2} are the standard deviations similarly of the existing kernel. Finally, these values are used to compute a similarity score s , given by,

$$s = \max(0, 1 - \frac{d^2}{1.5 \min(\sigma_1^2, \sigma_2^2)}). \quad (3)$$

When determining the similarity score, we wish to consider not only the separation of the centres of the kernels, but also their respective spreads. The sizes and shapes of the distributions affect the similarity score, as well as the locations of their centres. The existing kernel with the largest similarity score is used as the kernel of the tracked object in the local camera if the squared distance, d^2 , is less than an empirically determined threshold, e.g. $\frac{\min(\sigma_1^2, \sigma_2^2)}{6}$. Otherwise, the new kernel is constructed and used.

Linking kernels using the signature-based kernel matching method.

After constructing and disseminating the local track snapshot and receiving foreign track snapshots to and from other cameras respectively, links between the local kernel and kernels in foreign cameras are determined using a signature-based kernel matching method. This is performed as follows.

For each stored foreign track snapshot, the *visual distance* is found between the signature in the local track snapshot and the signature in the foreign track snapshot. In order to calculate this, first the signature in the local track snapshot is rescaled using a lighting compensation method. The histogram bin boundaries of the signature are linearly scaled by the ratio of the average luminance of the foreign signature to the average luminance of the local signature. Following this, using the distance metric defined on the signature, the visual distance between the brightness rescaled signature in the local track snapshot and the signature in the foreign track snapshot, d^s , is calculated. If the visual distance is greater than the maximum signature distance, d_{\max}^s , then no further processing is performed for that foreign track snapshot. Otherwise, the *signature weight* w^s is calculated:

$$w^s = d_{\max}^s - d^s \quad \text{if } d^s \leq d_{\max}^s. \quad (4)$$

The signature weight is then used to initialise or strengthen a linked pair of kernels between the local kernel and the kernel in the foreign track snapshot. If a link between these two kernels does not exist yet, one is created with a *kernel link weight*, w^k , equal to the signature weight, i.e.,

$$w_{\text{init}}^k = w^s. \quad (5)$$

Otherwise, if a linked pair of kernels already exists between these two kernels, its kernel link weight is incremented as follows,

$$w_{\text{new}}^k = w_{\text{old}}^k + w^s. \quad (6)$$

In addition, the transit time, t , given by,

$$t = f_{\text{local}} - f_{\text{foreign}} \quad (7)$$

is computed, where f_{local} and f_{foreign} are the frames in the local and foreign track snapshots respectively. This provides a measure of the time it took an object to move from the real-world location in the foreign camera to the real-world location in the local camera.

Each linked pair of kernels is associated with the most recent signature weight, w^s , the kernel link weight, w^k , and a history of transit times. The kernel link weight is used later in location-based kernel matching and when correspondences between tracks are found in section 2.3. The historical set of transit times are used later in computing temporal similarities in section 2.2. In addition, each linked pair of kernels is associated with a history of the local and foreign track snapshots each time the link was strengthened.

Linking kernels using the location-based kernel matching method.

Chan-ges in camera angle often affect the apparent pose of an object as viewed by the camera. As a result, different parts of the object may have different colours visible to different cameras. This leads to large signature distances when using the signature-based kernel matching method just described, which prevents kernels of tracks corresponding to the same real-world object from being correctly linked. The location-based kernel matching method described in this section addresses this problem. A history of linked pairs of kernels must have already been established before location-based kernel matching can be performed. This can be done by running the STAC algorithm using only signature-based kernel matching for some time before enabling location-based kernel matching.

The proposed location-based kernel matching is performed after receiving foreign track snapshots from other cameras, in parallel to the signature-based kernel matching method. For each received foreign track snapshot, of the set of *historical linked pairs of kernels* containing the kernel in the foreign track snapshot and a kernel in the local camera, the pair with the greatest kernel link weight is identified. A historical linked pair of kernels is any linked pair of kernels initialised in a previous frame. If the local track passed through the local kernel in this pair in a previous frame, then the current signature of the locally tracked object is replaced with its signature from this previous frame. Following this, the signature distance between the new local signature and the signature in the foreign track snapshot is calculated, and if it is below the threshold d_{max}^s , then the signature weight, kernel link weight and transit time are initialised or updated as per equations 4.7.

Selecting the best linked pairs of kernels. The kernel matching process described until this point will result in a set of linked pairs of kernels between local kernels that the locally tracked object passed through and kernels in foreign cameras. For a given locally tracked object, if there exists more than one linked pair of kernels between the local camera and a foreign camera, only the

linked pair of kernels with the greatest kernel link weight is selected for further processing. This is to try and prevent a track in the local camera from being linked to multiple tracks in a foreign camera. This results in a set of best linked pairs of kernels, up to one per foreign camera, for each locally tracked object. At this point, all initialisations or updates that were made to kernel link weights and transit time histories of linked pairs that were not selected, are discarded.

2.2 Finding Spatial and Temporal Similarities between Current Pairs of Objects and Historical Pairs of Objects

We now examine the similarity of the locations and transit times of each of the best linked pairs of kernels to the locations and transit times of historical linked pairs of kernels between the same two cameras. This gathers evidence that these locations, and hence the tracks of the objects appearing in them, have reliably shown over time to correspond to the same real world object. For each best linked pair of kernels, this process finds the *spatial similarity* and *temporal similarity* between the best linked pair of kernels and each historical linked pair of kernels between the same two cameras as the best linked pair of kernels. For simplicity, we describe the process for one best linked pair of kernels, which we will call the *current linked pair of kernels*. Fig. 2 illustrates an example of the current linked pair of kernels and two relevant historical linked pairs of kernels.

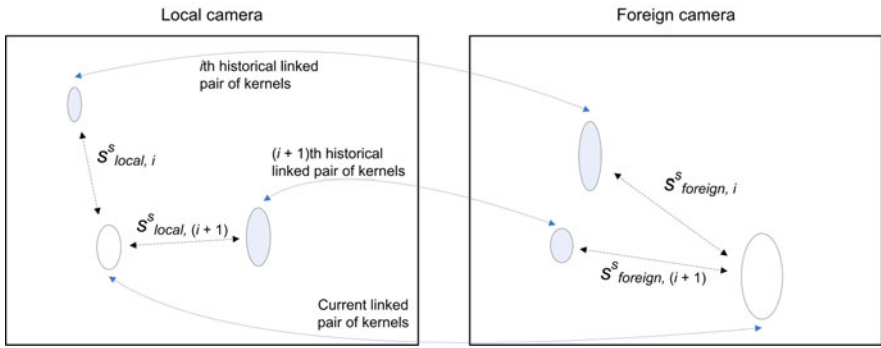


Fig. 2. A diagram showing how the spatial similarity is calculated between kernels in the current and each historical linked pair of kernels. In this diagram, ellipses represent the distributions of the kernels at one standard deviation from the mean.

For each historical linked pair of kernels, two spatial similarities are calculated, as shown in Fig. 2. The spatial similarity, $s^s_{\text{local},i}$ of the local kernel in the current linked pair of kernels and the local kernel in the i th historical linked pair of kernels is given by,

$$s^s_{\text{local},i} = \exp \left(- \left(\frac{\Delta x_{\text{local},i}^2}{\sigma_{x_{\text{local},i}}^2} + \frac{\Delta y_{\text{local},i}^2}{\sigma_{y_{\text{local},i}}^2} \right) / 2 \right) \quad (8)$$

where $\Delta x_{\text{local},i}$ and $\Delta y_{\text{local},i}$ are the difference in the x - and y -coordinates of the two local kernel centres respectively, and $\sigma_{x_{\text{local},i}}$ and $\sigma_{y_{\text{local},i}}$ are the standard deviations of the local kernel in the i th historical linked pair of kernels. A second spatial similarity, $s_{\text{foreign},i}^s$, is similarly computed for the two foreign kernels.

A temporal similarity, s_i^t , between the current and i th historical linked pairs of kernels is computed and is given by,

$$s_i^t = \begin{cases} \exp\left(-\frac{(t-\bar{t})^2}{\sigma_t^2}\right), & \sigma_t \neq 0 \\ 1, & \sigma_t = 0 \end{cases}. \quad (9)$$

where t is the most recent transit time associated with the current linked pair of kernels, and \bar{t} and σ_t are the mean and standard deviation respectively of the set of transit times associated with the historical linked pair of kernels.

2.3 Determining Correspondences between Pairs of Tracks

Based on the spatial and temporal similarities computed so far for the current linked pair of kernels, a *track link weight* is calculated. This represents the likelihood that the local and foreign tracks in the most recent track snapshots associated with the current linked pair of kernels represent the same real-world object. This link is represented by the track IDs of the respective tracks. If a track link weight has not yet been initialised between these two tracks, then it is initialised as follows,

$$w_{\text{init}}^{\text{tr}} = w^s \sum_i s_{\text{local},i}^s \cdot s_{\text{foreign},i}^s \cdot s_i^t \cdot w_i^k, \quad (10)$$

and if a track link weight already exists between the two tracks, then it is incremented as follows,

$$w_{\text{init}}^{\text{tr}} = w_{\text{old}}^{\text{tr}} + w^s \sum_i s_{\text{local},i}^s \cdot s_{\text{foreign},i}^s \cdot s_i^t \cdot w_i^k, \quad (11)$$

where w_i^k is the kernel link weight of the i th historical linked pair of kernels. By summing the spatial and temporal similarities, we effectively assemble a weighted kernel density estimator, where each weight is given by a kernel link weight. In addition, a track link counter, m , of the number of times the track link weight has been increased, is kept. The track link counter provides a measure of the consistency of the evidence used to determine the track link weight.

As visual, spatial and temporal evidence is accumulated that these two tracks represent the same real world object, their track link weight and track link counter increases. Once both are sufficiently large, we can be confident there is a *correspondence* between the tracks. Specifically, a correspondence between two tracks is declared if in any frame their track link weight crosses a first threshold, $w_{\text{min}}^{\text{tr}}$, and if their track link count crosses a second threshold, m_{min} , i.e.,

$$C = \begin{cases} \text{True}, & \text{if } w^{\text{tr}} > w_{\text{min}}^{\text{tr}} \text{ and } m > m_{\text{min}} \\ \text{False}, & \text{otherwise} \end{cases} \quad (12)$$

where C is a Boolean variable representing if there exists a correspondence between the tracks whose track link weight is w^{tr} and track link count is m . If insufficient evidence has been collected then the track link weight and track link count may not satisfy these conditions and the correspondence will not be found. This will be the case when the camera network is first initialised and may also happen for very short tracks.

Additionally, correspondences between tracks are declared following feedback from foreign cameras. When a camera declares a correspondence between a pair of tracks using equation 12, this correspondence is broadcast to all other cameras. If this information implies that a track in the local camera and a track in a foreign camera are the same real-world object, then a correspondence is declared between these two tracks. This prevents asymmetry in the correspondences across cameras, which would otherwise result from the fact that each camera maintains its own set of tracking statistics.

Once a correspondence is found between two tracks, for objects from these tracks that appear in future frames, the condition $d^s \leq d_{\max}^s$ in equation 4 need not hold for a linked pair of kernels to be initialised or strengthened between their kernels. This ensures that statistics relating to the kernels in these two tracks continue to be collected, for use in further multi-camera tracking.

3 Evaluation

3.1 Evaluation Datasets

Seven multi-camera video sequences were used as test sets for evaluation. The evaluation test sets contained two or four cameras and comprised common FOV scenarios, disjoint FOV scenarios, and hybrid scenarios containing some overlapping and some disjoint camera views. Test sets were recorded in an indoor office environment with uncontrolled lighting. Videos ranged in length from 2 minutes 50 seconds to 10 minutes 30 seconds, at approximately 10 frames per second (fps) and at a resolution of 768×576 pixels or of 640×480 pixels.

3.2 Experimental Setup

Kalman Filter based single camera object tracking results were sanitised manually and used as input to the multi-camera system. Short tracks were automatically removed as noise, even if they were not, for consistency.

The signature used in our evaluations divides the bounding box of an object into a 4×4 grid and calculates luminance and hue histograms for the pixels in each cell in this grid. Each histogram contains 8 bins, giving a total of 256 bins. Euclidean distance was used to calculate the signature distance in section 2.1. We selected this signature type as it is used by our single-camera tracking algorithm in our overall object detection and tracking framework. However, it should be noted that the multi-camera tracking algorithm presented in this paper could be used with any signature type and distance metric.

The values used for each parameter of the STAC algorithm during evaluation were: $d_{\max}^s = 2.0$; $w_{\min}^{tr} = 1.0$; $m_{\min} = 20$. These values were selected empirically, and based on experience with our particular signature type. Additionally, a maximum of 100 foreign track snapshots were stored by a camera at any time.

3.3 Evaluation Metrics

We considered two metrics for our evaluation. The F_1 score measures the robustness of results. The correspondence delay (CD) measures how long it takes to link tracks. A higher F_1 score and a lower CD are desirable.

The F_1 score is the harmonic mean of recall and precision, balancing the trade-off between false positives and false negatives. Recall and precision have been used previously in evaluating tracking-based events [8]. Here, we treat a correct correspondence as a true positive, an incorrect correspondence as a false positive and the lack of a correspondence where one should have been found as a false negative. The F_1 score is given by,

$$F_1 = 2 \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \quad (13)$$

For a given correspondence, the *correspondence delay* (CD) is the time between the second track becoming visible and a correspondence being found between the two tracks. The CD for a test set is the mean of the CDs for the true positive correspondences for that test set. To our knowledge, the timeliness of correspondences has not been considered in the literature before.

4 Results and Discussion

Table 1 shows a summary of the results of evaluating the STAC algorithm across all test sets for the two cases of:

- the STAC algorithm described in section 2 but without using the location-based kernel matching method described in section 2.1; and
- the STAC algorithm with the use of location-based kernel matching.

Table 1. Overall results of evaluating the STAC algorithm with and without location-based kernel matching (LKM) across all seven test sets

Metric	Without LKM	With LKM
Recall	0.54	0.66
Precision	0.63	0.68
F_1 score	0.58	0.67
CD (frames)	31.2	21.7

Table 1 shows that the addition of location-based kernel matching improves the overall accuracy and speed of the multi-camera tracking system, with a particularly marked increase in recall and decrease in CD. Generally, location-based kernel matching selects a local signature better matching the foreign track signature only if the tracks are a true correspondence, i.e., only if they represent the same real-world object. This is expected to increase true positives without increasing false positives. Importantly, location-based kernel matching is not designed to select a signature that is a worse match if the tracks are not a true correspondence, which would result in fewer false positives. These expectations are reflected in the relatively stronger increase in recall than precision, and no decrease in either. Additionally, since signature distances for true correspondences are reduced when using location-based kernel matching, it is expected that the condition in equation 4 will be satisfied more often and the signature weights will be greater, thus the conditions in equation 12 will be satisfied after fewer frames. This is reflected in the reduced CD.

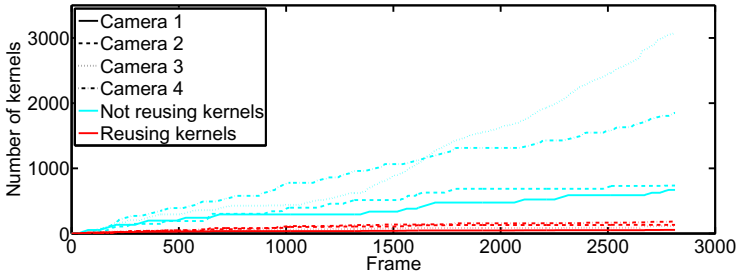


Fig. 3. The number of kernels stored for a test case when new kernels are created for every observed object, and when an existing kernel is reused that is an approximation for the size, shape and location the observed object using the method described in section 2.1. Reusing kernels reduces memory usage and achieves better convergence.

Fig. 3 demonstrates a comparison between the number of kernels stored by the STAC algorithm for each camera over time when kernels are created for every observed object in each camera, and for when kernels are reused using the method described in section 2.1. In Fig. 3, for the case of not reusing kernels, we observe a roughly linear increase in the number of kernels stored, as expected. In contrast, Fig. 3 shows that the number of kernels stored is bounded when kernels are reused. These results imply a similar result for the number of historical linked pairs of kernels stored. This suggests that when reusing kernels, a guarantee can be made on the memory usage of the system. Alternatively, a memory limit can be placed on the system without compromising tracking accuracy. This result is particularly important for embedded applications with restricted memory resources. Although for brevity results are shown only for one test set, these trends are reflected in other test sets.

5 Conclusions

The Signature-based Tracking Across Cameras (STAC) algorithm as part of a distributed tracking framework enables real-time multi-camera tracking without a training phase. The kernel-based tracking algorithm covers the entire field of view of each camera rather than only entry and exit points, and continuously collects and updates tracking statistics. Reusing kernels enables the collection of tracking statistics. Also, reusing kernels places a bound on memory usage, allowing implementation in an embedded application. The novel location-based kernel matching method uses tracking statistics to accommodate abrupt and unpredictable changes in the visual characteristics of objects within and across camera views. We showed that STAC's tracking accuracy and speed were improved over seven test sets by the addition of location-based kernel matching.

References

1. Arsic, D., Lyutskanov, A., Rigoll, G., Kwolek, B.: Multi camera person tracking applying a graph-cuts based foreground segmentation in a homography framework. In: 12th IEEE Int Workshop on Performance Evaluation of Tracking and Surveillance (PETS-Winter), pp. 1–8. IEEE, Los Alamitos (2009)
2. Khan, S., Shah, M.: Consistent labeling of tracked objects in multiple cameras with overlapping fields of view. *IEEE Trans. Pattern Anal. Mach. Intell.* 25, 1355–1360 (2003)
3. Lee, L., Romano, R., Stein, G.: Monitoring activities from multiple video streams: establishing a common coordinate frame. *IEEE Trans. Pattern Anal. Mach. Intell.* 22, 758–767 (2000)
4. Makris, D., Ellis, T., Black, J.: Bridging the gaps between cameras. In: Int Conf Computer Vision Pattern Recognition (CVPR), pp. 205–210. IEEE, Los Alamitos (2004)
5. Cichowski, A., Madden, C., Detmold, H., Dick, A., van den Hengel, A., Hill, R.: Tracking hand-off in large surveillance networks. In: 24th Int Conf on Image and Vision Computing New Zealand (IVCNZ), pp. 276–281. IEEE, Los Alamitos (2009)
6. Sheikh, Y., Li, X., Shah, M.: Trajectory Association across Non-overlapping Moving Cameras in Planar Scenes. In: Int Conf Computer Vision Pattern Recognition (CVPR), pp. 1–7. IEEE, Los Alamitos (2007)
7. Javed, O., Rasheed, Z., Shafique, K., Shah, M.: Tracking across multiple cameras with disjoint views. In: 9th IEEE Int Conf on Computer Vision (ICCV), pp. 952–957. IEEE, Los Alamitos (2003)
8. Petrushin, V.A., Wei, G., Gershman, A.V.: Multiple-camera people localization in an indoor environment. *Knowl Inf. Syst.* 10, 229–241 (2006)

A Template Matching and Ellipse Modeling Approach to Detecting Lane Markers

Amol Borkar, Monson Hayes, and Mark T. Smith

Georgia Institute of Technology, Atlanta, GA, USA
{amol,mhh3}@gatech.edu
Kungliga Tekniska Högskolan, Stockholm, Sweden
{msmith}@kth.se

Abstract. Lane detection is an important element of most driver assistance applications. A new lane detection technique that is able to withstand some of the common issues like illumination changes, surface irregularities, scattered shadows, and presence of neighboring vehicles is presented in this paper. At first, inverse perspective mapping and color space conversion is performed on the input image. Then, the images are cross-correlated with a collection of predefined templates to find candidate lane regions. These regions then undergo connected components analysis, morphological operations, and elliptical projections to approximate positions of the lane markers. The implementation of the Kalman filter enables tracking lane markers on curved roads while RANSAC helps improve estimates by eliminating outliers. Finally, a new method for calculating errors between the detected lane markers and ground truth is presented. The developed system showed good performance when tested with real-world driving videos containing variations in illumination, road surface, and traffic conditions.

Keywords: Lane Detection and Lane Keeping, Template Matching, Driver Assistance Systems, Advanced Vehicle Safety Systems.

1 Introduction

Driver safety has always been an area of interest to automotive research. With the advancement of semiconductor design, powerful electronic devices with small footprints are starting to appear in many vehicles. These devices are capable of performing various tasks to assist the driver of an automobile paving the way for Driver Assistance (DA) systems.

One of the many task performed by such a DA system is Lane Departure Warning (LDW). In LDW, the positions of lane markers around the host vehicle are continuously monitored to determine if a lane change is imminent with the help of exogenous inputs like steering angle, commuting speed, and rate of lane marker movement. Consequently, a vital component of LDW is lane detection which is described as a problem of locating painted white or yellow markings on the road surface. In vision based lane detectors, a camera mounted under the rear-view mirror is used to acquire data for lane detection.

This paper is organized as follows: subsequent to the introduction, a brief literature review is conducted citing some of the current lane detection implementations. Then, the different components used in the detection and tracking of lane markers are explained. Finally, the method for calculating errors is described. The performance of the proposed system is assessed on real world videos recorded at various times of the day. Finally, the conclusion and planned improvements are discussed.

2 Prior Research

Lane detection is still an active area of automotive research. Conventional approaches suggest the application of thresholds after studying patterns in histograms in hopes of segmenting lane marker pixels from background or road pixels [1,2]. Unfortunately, histogram approaches are vulnerable to outlier intensity spikes. The use of edge images to find lines or curves using a variety of kernel operators has been suggested by [3,4,5,6] but face difficulty when markers show signs of age and wear. A piece-wise Hough transform to fit a line on a curve has been used to handle conditions involving scattered shadows [7,8]. Additionally, the incorporation of edge directions has been used to remove some false signalling [2,6]. Unfortunately, invariance to scale and rotation tends to be major problem for these methods. Classifying small image blocks as lane markers using learning methods has been suggested by [9]. But a good quality linear classifier is difficult to derive without an infinitely large catalog of negative training examples. Lane detection using adaptive thresholds and one dimensional iterated matched filters has been suggested by [10,11]. Unfortunately, one dimensional template matching did not perform so well during the day.

Lane detection is a crucial component of many DA systems; thus, it needs to be extremely reliable and robust. Current research appears to boast high performance only in the presence of favorable illumination and road surface conditions. Unfortunately, these conditions are unlikely to exist on the road network in most big cities. Based on the literature survey, it can be seen the feature extraction stages in existing implementations are unable to satisfactorily discriminate between lane markers and surface artifacts. Consequently, there is a need to develop improved techniques to detect lane markers that is able to cope with the variety of road conditions that exist around the world.

3 System Overview

The overview of the proposed lane detector is shown in Fig. 1. First, the images undergo preprocessing the form of Inverse Perspective Mapping (IPM) and color conversion. Then, template matching in addition to morphology and ellipse projection finds areas containing lane markers. Finally, the Kalman filter is used to track lane marker estimates while RANSAC helps eliminate outliers.

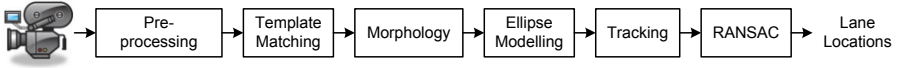


Fig. 1. System overview

4 Lane Marker Detection

4.1 Preprocessing

Inverse Perspective Mapping (IPM) is used to convert a camera perspective image to a bird’s-eye view of the scene. The transformation given by

$$X(r) = h \cdot \left(\frac{1 + \left[1 - 2 \left(\frac{r-1}{M-1} \right) \right] \tan \alpha_v \tan \theta_o}{\tan \theta_o - \left[1 - 2 \left(\frac{r-1}{M-1} \right) \right] \tan \alpha_v} \right) \tag{1}$$

$$Y(r, c) = h \cdot \left(\frac{\left[1 - 2 \left(\frac{c-1}{N-1} \right) \right] \tan \alpha_u}{\sin \theta_o - \left[1 - 2 \left(\frac{r-1}{M-1} \right) \right] \tan \alpha_v \cos \theta_o} \right) \tag{2}$$

uses camera calibration parameters such as height from the ground (h), vertical field of view (α_v), horizontal field of view (α_u), tilt angle below the horizon (θ_o), and mage dimensions ($M \times N$) to map pixels from the image plane to the world [11]. The transformed image is converted from RGB to YCbCr to aid color segmentation [12].

4.2 Template Matching

Specific dimensions for different lane markings have been defined by the Federal Highway Administration (FHA). Normal and wide lane markers are approx. 6 inches and 10 inches wide respectively. Double lane markers consist of two normal lane markers with a gap in between. [13]. Templates shown in Fig. 2 are created with equivalent dimensions in the IPM image and used for matching.

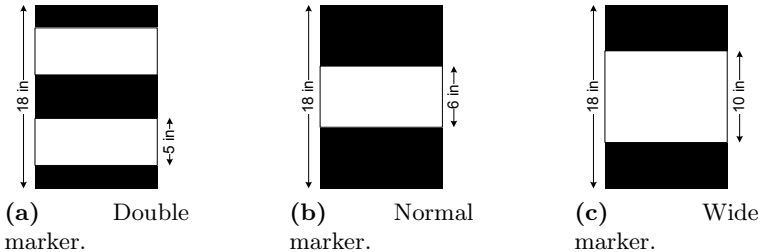


Fig. 2. Lane marker templates and their dimensions

Template matching is performed using Normalized Cross Correlation (NCC). At first, a binary image is obtained by application of a high threshold τ_{High} on

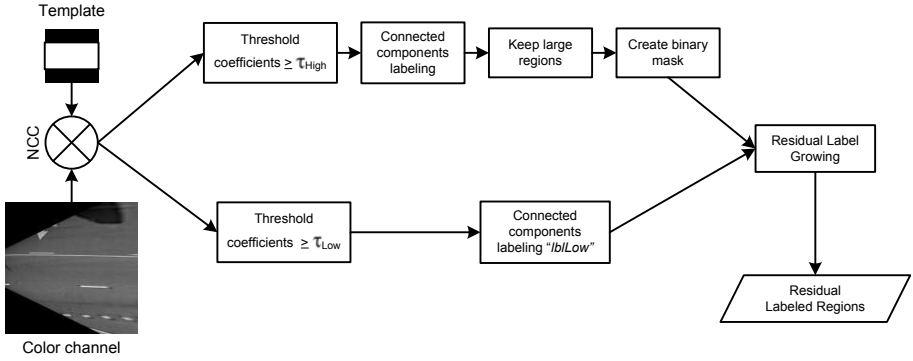


Fig. 3. Flowchart illustrating template matching procedure

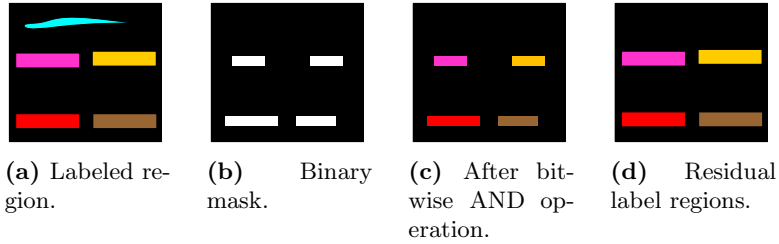


Fig. 4. Steps involved in the Residual Label Growing

the coefficients map. The resultant binary image undergoes connected components labeling as a way to ignore small regions while keeping large connected regions. These remaining labels will serve as a binary mask. Another application of connected components labeling is used to create a separate labeled region called “lblLow” from the coefficients map thresholded by τ_{Low} . Finally, Residual Label Growing (RLG) is used to find residual labeled regions which will be sent for morphological processing.

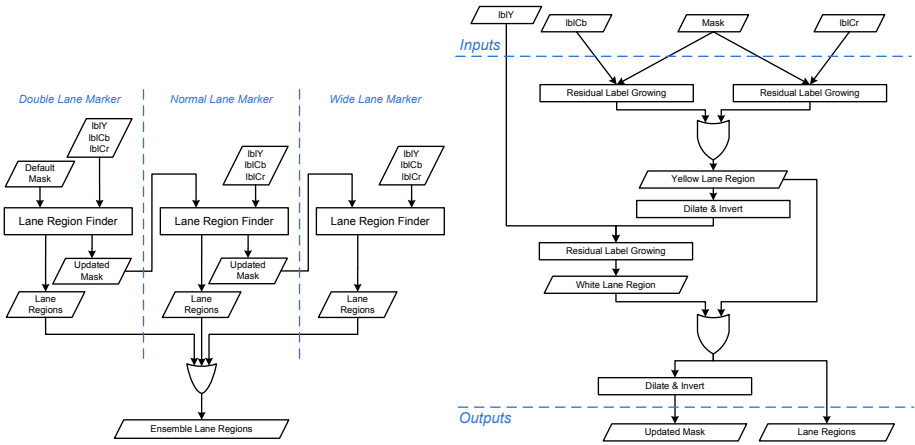
The Residual Label Growing process can be explained by referring to Fig. 4. A labeled region containing lane markers and a stray object are shown in Fig. 4a. Each color in Fig. 4a represents a different label with black being the background. A bitwise AND operation is performed between the labeled region and the binary to mask in Fig. 4b to produce residual labels as shown in Fig. 4c. Consequently, the entire regions corresponding to the residual labels are extracted and serve as the desired output as shown in Fig. 4d. The RLG process uses a hysteresis approach to segment lane markers by utilizing a high and low threshold on the coefficients map. It is evident by comparing Fig. 4a and 4c that the stray object shown in cyan has been eliminated while the lane markers are untouched. With a mask created using an appropriate τ_{High} threshold, similar stray objects can be ignored. This is a key feature of the RLG.

The template matching procedure for the Y channel with a normal lane marker template is illustrated in Fig. 3. To detect the other markers on the road that

vary in shape and color, the template matching procedure is repeated using the remaining combination of templates and color channels.

4.3 Morphology

The residual labeled regions acquired after template matching are fed as inputs to the Lane Region Finder (LRF) as shown in Fig. 5a. The labeled regions of the Y, Cb, and Cr channels are prefixed with “lbl” e.g. lblY, lblCb, and lblCr. The default mask passes all labels untouched (all-pass). After each LRF iteration, the mask is updated to ignore areas that have already been detected. This updated mask is then fed to the next LRF module. Finally, the results of all three LRFs are combined using a bitwise OR operation to produce Ensemble Lane Regions (ELR).



(a) Flowchart illustrating the creation of ensemble lane regions. (b) Flowchart illustrating the inner workings of the Lane Region Finder (LRF).

Fig. 5. Steps involved in the Morphology

The ordering of input labels plays an important role in the success of the lane detector as the mask used in LRF is updated after each operation. When an off-line test involving the computation of NCC coefficients between equal sized test samples and their corresponding templates was conducted, on average the double lane templates produced the highest cross correlation score in the center of the coefficients map. This was followed by the scores of narrow and then wide lane templates. The double lane templates were able to produce a higher correlation score as multiple intensity oscillations in the vertical direction account for strong discriminating features in comparison to the other templates. As a result, the LRF sequencing in Fig. 5a starts by detecting double lane markers. The updated mask is used in the LRF operation in collaboration with residuals labels corresponding to normal lane markers, and then with wide lane markers.

The inner workings of the Lane Region Finder is shown in Fig. 5b. First, two Residual Label Growing (RLG) operations are performed using a binary mask; one on $lblCb$ and the other on $lblCr$. The two residual labeled regions are converted to binary images by assigning each non-background label to a binary one. The two binary images are combined using a bitwise OR operation and assigned as yellow lane region. This region is then dilated and inverted to serve as mask. The purpose of the mask is to avoid re-detection in areas that have already been considered as yellow lane regions. The structuring element used in dilation is row vector with a height equivalent to 12ft in the IPM image (6ft on either side of the detected yellow regions). RLG is performed again but this time on $lblY$ using the dilated and inverted mask. The white and yellow marker regions are merged to depict detected lane regions corresponding to a particular template. The merged region is also inverted and dilated to serve as a mask for the next LRF block in Fig. 5a.

Yellow color content is prevalent in the Cb and Cr channels of the $YCbCr$ space. Depending on the intensity of yellow, the Y channel may occasionally contribute some information. As a result, yellow lane marker detection is performed using Cb and Cr components. However, white color is intensity dependent and contributed to only by the Y channel. Hence, white lane marker detection is performed using only the Y components.

4.4 Ellipse Modeling

Connected components labeling is performed on the Ensemble Lane Regions (ELR) in Fig. 5a to find a collection of distinct objects. The horizontal lengths of each object is measured for classification. Based on the FHA specifications, if the measured length of an object exceeds the equivalent of 10ft in the IPM image, then it is categorized as a full or solid line. Otherwise, the object is categorized as a broken line. In Fig. 6, the solid line is shown in red and broken lines are shown in Cyan. Each broken line is modeled as an ellipse whose major axis shown in brown is projected towards the front of the vehicle on the left. For solid lines, the leading 10ft of pixels shown in yellow are used in modeling the ellipse. The major axis of the ellipse allows to estimate the traversing direction of the markers from one frame to the next. The Hough transform was initially used to approximate

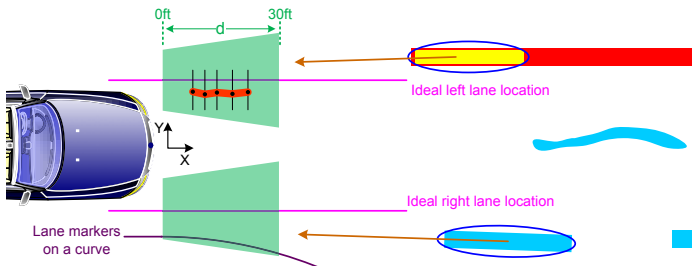


Fig. 6. Ellipse modeling and major axis projection

this direction; however, the path containing maximum pixels was often chosen but incorrect.

Given the FHA specifications of lanes being 12ft wide [13] and assuming that the vehicle is traveling in the middle of a straight road, the expected ideal locations of left and right lane marker centers are shown in pink. The major axis projection of an object is expected to be within the trapezoid at 0ft and 30ft to be considered as either a left or right lane marker candidate. The axis of symmetry of the trapezoid is aligned with ideal lane marker locations with the short base set to a length of 8ft, i.e. 4ft on either side of the ideal lane location in the IPM image. The length of the long base is set to entirely accommodate a circular arc within the 0-30ft range. This arc is assumed to represent lane markers on a curve and is shown in purple in Fig. 6. The radius of curvature of the arc is set to 65ft which is the American Association of State Highway and Transportation Official's (AASHTO) recommendations for minimum radius of curvature for a horizontal road curve with $e=4.0\%$ superelevation when traveling at speed of 20mph [14]. An isosceles trapezoid is chosen as the shape of the green window as opposed to a rectangle or triangle to allow detection of lane markers on a curve that may be offset from the ideal lane locations while at the same time reduce detection of artifacts or other markers far away from the vehicle. In the case that a solid or broken line exists inside the trapezoid as shown by the orange object, the pixel locations of the object are sampled at one foot intervals shown by the vertical black lines in Fig. 6 within the 0-30ft range. The black dots in each interval will serve as the measurements for the Kalman filter (see next section).

5 Tracking

The Kalman filter is used to estimate the lane marker movements from one frame to next. Measurements are acquired by sampling the object at one foot intervals inside the trapezoid as described earlier; consequently, separate Kalman filters are evaluated at every interval for both left and right lane markers. The state vector and corresponding equations are set as

$$\mathbf{x}(n) = [x(n) \ \dot{x}(n)]^T \quad (3)$$

$$\mathbf{x}(n+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \mathbf{x}(n) + \begin{bmatrix} N(0, \sigma_{w1}) \\ N(0, \sigma_{w2}) \end{bmatrix} \quad (4)$$

$$\mathbf{y}(n) = [1 \ 0] \mathbf{x}(n) + N(0, \sigma_v) \quad (5)$$

where $x(n)$ is the position or y-value and $\dot{x}(n)$ is the velocity of the lane marker in each interval. The noise in the state and measurement equations is assumed to be white and each process is assumed to be uncorrelated with the others. After initialization, if no measurement is made at a particular interval, the Kalman filter relies on its prediction to produce the estimate. However, after 50 sequential predictions, it is deactivated at that particular interval to avoid producing

estimates as lane markers may not actually exist. Separate Kalman filters are evaluated at every interval rather than collectively in one matrix to avoid the unsolvable condition where the prediction counter exceeds 50 and the Kalman filters have been deactivated at certain intervals .

6 Outlier Elimination

The estimates produced by the Kalman filter undergo Random Sample Consensus (RANSAC) to eliminate outliers as shown in Fig. 7a. Normally, k-RANSAC or quadratic RANSAC would be used in outlier elimination for fitting a curve [15]; however, they are computationally intensive and slow. Luckily, since the minimum radius of curvature recommended by AASHTO [14] is large, inlier estimation using a straight line model in RANSAC with an appropriate error threshold is sufficient. This threshold is calculated using simple properties of a circle.

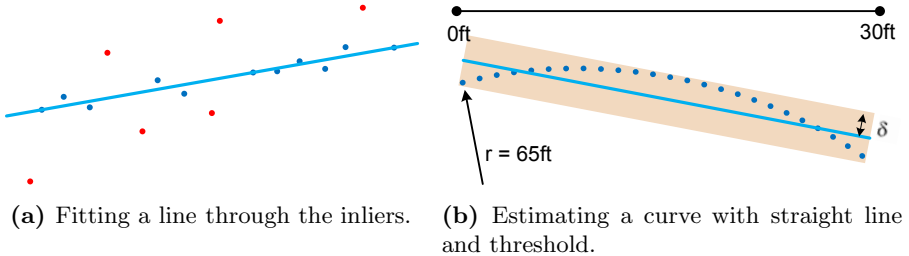


Fig. 7. RANSAC for inlier estimation

If lane markers lie on a curve with radius 65ft [14], this curve can be viewed as an arc of a circle with the same radius as represented by the dotted blue line in Fig. 7b. A circular segment can be created by joining the end points of this arc. $\frac{1}{2}$ of the height of the circular segment is the minimum error threshold allowing the ideal line model in RANSAC to contain all the points along the curve. The ideal line is shown in cyan and the threshold (δ) is given by

$$\delta = \frac{r - \sqrt{r^2 - \frac{(r - \sqrt{r^2 - d^2})^2 + d^2}{4}}}{2} \tag{6}$$

Finally, Ordinary Least Squares (OLS) estimation is used to fit a quadratic curve on the remaining inliers. Each dot in the dotted blue line in Fig. 7b is an estimate produced by the Kalman filter at one foot intervals within the 0-30ft range.

7 Error Estimation

First, the ground truth is generated using the *Time-Slice* approach which allows to quickly and accurately annotate videos [16]. The error in each frame

is then computed by determining the maximum distance between the detected lane marker locations and that of the ground truth. The ground truth data is also transformed to the IPM domain using Eq. (1) and (2) to allow the accurate computation of these distances. In the bird's-eye view, the inter-pixel distances have linear correspondences in the world; as a result, the distances computed in any portion of the image can be easily mapped to a physical distance.

The distances are computed at one foot intervals up to 30ft ahead of the vehicle. The error is determined by calculating

$$\lambda_{(i,f)} = \max(|Gt_{(i,f)} - X_{(i,f)}| - \frac{W}{2}, 0) \text{ s.t. } i \in [0, 30] \tag{7}$$

$$E(f) = \|\lambda\|_\infty = \max_i \lambda_{(i,f)} \tag{8}$$

where $Gt_{(i,f)}$ is the ground truth location of the lane marker and $X_{(i,f)}$ is the detected lane location in frame f at a distance of i feet ahead of the car. W is an interval around the ground truth locations and is set to the equivalent of 8 inches in the IPM image. This value is chosen as the mean of the widths of normal and wide lane markers based on the specifications of the Federal Highway Administration (FHA) [13]. Consequently, lane marker estimates that fall within the interval specified by W are categorized as having no error. As a result, the error in each frame, $E(f)$ is computed as the L-Infinity Norm of the λ values. This idea is illustrated in Fig. 8 where the green line marks the ground truth, the blue line is the lane marker estimation using the proposed lane detector, and $\lambda_{(i,f)}$ is the offset measured at specific distances ahead of the vehicle.

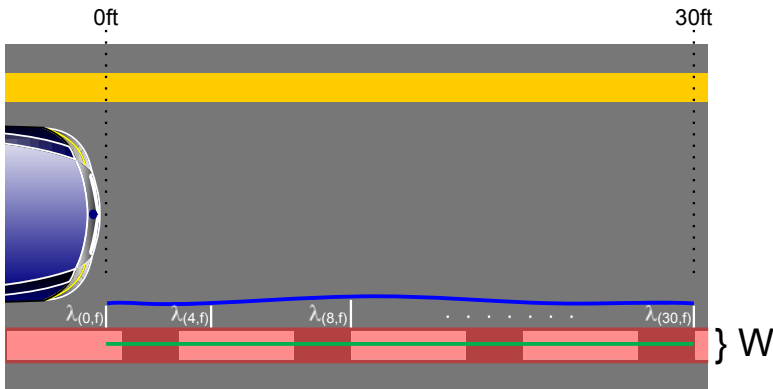


Fig. 8. Calculating errors using λ distances

8 Results and Analysis

The following rules were used to quantify the results into the different categories:

1. A correct detection occurs when less than $\frac{N}{2}$ λ distances are greater than 0.
2. A missed detection occurs when more than $\frac{N}{2}$ λ distances are greater than 0.

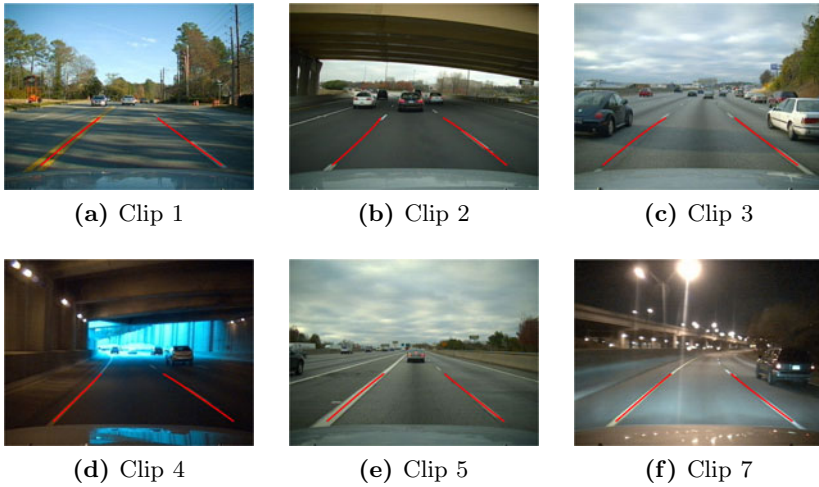


Fig. 9. Scenes from video clips used in testing with correct lane detections

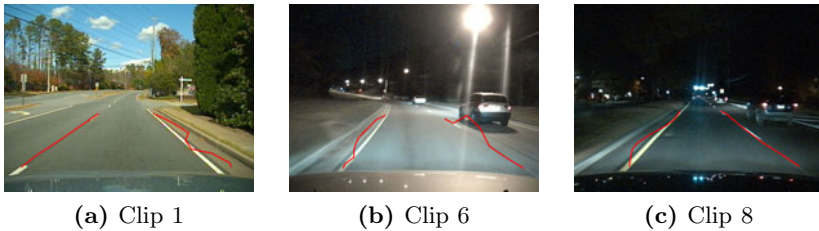


Fig. 10. Few examples of missed detections

N is the number of computed distances for error calculation (i.e. $N = 30$). Eight ten minute long video clips were used to evaluate the lane detector. Each video clip was annotated ahead of time using the *Time-Slice* approach [16]. The errors represented by $E(f)$ were calculated only when an estimate was considered correct. Table 1 presents a quantitative evaluation of the proposed system. Methods of template matching, morphology, and ellipse modeling allows the lane detector to successfully deal with most issues associated with scattered shadows, illumination changes, surface irregularities, and presence of vehicles in neighboring lanes. In addition, the Kalman filter and RANSAC enables detection and tracking of markers on curved and winding roads which was initially problematic. A few scenes from the test clips with correct lane detections are shown in Fig. 9.

A few instances of missed detections are also shown in Fig. 10. Missed detections were most commonly caused by lens flares from overhead streetlights and pavements running parallel to the road. The reason behind this is that both of these objects produced shapes in the IPM image that closely resembled the lane marker templates often leading to their detections.

Table 1. Accuracy of the proposed lane detection system

	Left Lane		Right Lane	
	Correct	Avg. E(f) ft.	Correct	Avg. E(f) ft.
Clip 1	96.31 %	0.018	97.21 %	0.035
Clip 2	95.92 %	0.033	96.94 %	0.024
Clip 3	94.26 %	0.012	95.50 %	0.020
Clip 4	82.08 %	0.012	94.02 %	0.019
Clip 5	100 %	0	99.48 %	0.012
Clip 6	95.78 %	0.014	99.53 %	0.020
Clip 7	100 %	0	100 %	0
Clip 8	73.28 %	0.019	96.50 %	0.032

9 Conclusion

A new lane detection system is presented in this paper. At first, the input image undergoes a geometric transformation followed by a color space conversion. Then the procedures for detecting lane markers using template matching, morphology, and elliptical modeling are explained. Kalman filtering and RANSAC used in tracking and outlier elimination greatly helps in handling lane marker extraction on curves. Finally, a new technique to calculate lane detection errors is also introduced. Despite the presence of scattered shadows, illumination changes, surface irregularities, and vehicles in neighboring lanes, our proposed system showed very good performance which is portrayed by the quantitative results in Table 1.

10 Future Work

The application of constraints on the width between the detected lane markers is being explored. This should help prevent the sporadic oscillations of lane marker estimates.

References

1. Lim, K., Seng, K., Ang, L., Chin, S.: Lane Detection and Kalman-Based Linear-Parabolic Lane Tracking. In: 2009 International Conference on Intelligent Human-Machine Systems and Cybernetics, pp. 351–354. IEEE, Los Alamitos (2009)
2. Lee, J.-W., Cho, J.-S.: Effective lane detection and tracking method using statistical modeling of color and lane edge-orientation. In: Fourth International Conference on Computer Sciences and Convergence Information Technology, ICCIT 2009, pp. 1586–1591 (2009)
3. Wu, H., Liu, Y., Rong, J.: A lane detection approach for driver assistance. In: International Conference on Information Engineering and Computer Science, ICIECS 2009, pp. 1–4 (December 2009)

4. Haselhoff, A., Kummert, A.: 2d line filters for vision-based lane detection and tracking. In: International Workshop on Multidimensional (nD) Systems, nDS 2009, July 1–29, pp. 1–5 (2009)
5. Li, S., Shimomura, Y.: Lane marking detection by side Fisheye Camera. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2008, pp. 606–611 (2008)
6. Gong, J., Wang, A., Zhai, Y., Xiong, G., Zhou, P., Chen, H.: High speed lane recognition under complex road conditions. In: 2008 IEEE Intelligent Vehicles Symposium, pp. 566–570 (2008)
7. Taoka, T., Manabe, M., Fukui, M.: An Efficient Curvature Lane Recognition Algorithm by Piecewise Linear Approach. In: IEEE 65th Vehicular Technology Conference, VTC 2007-Spring, pp. 2530–2534 (2007)
8. Truong, Q., Lee, B., Heo, N., Yum, Y., Kim, J.: Lane boundaries detection algorithm using vector lane concept. In: 10th International Conference on Control, Automation, Robotics and Vision, ICARCV 2008, pp. 2319–2325 (2008)
9. Kim, Z.: Robust lane detection and tracking in challenging scenarios. *IEEE Transactions on Intelligent Transportation Systems* 9(1), 16–26 (2008)
10. Borkar, A., Hayes, M., Smith, M.: Robust lane detection and tracking with ransac and kalman filter. In: 2009 IEEE International Conference on Image Processing (2009)
11. Borkar, A., Hayes, M., Smith, M.: Lane Detection and Tracking Using a Layered Approach. In: Blanc-Talon, J., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2009. LNCS, vol. 5807, pp. 474–484. Springer, Heidelberg (2009)
12. International Telecommunication Union: Rec. ITU-R BT.601-5: Studio Encoding Parameters of Digital Television for Standard 4:3 and Wide-screen 16:9 Aspect Ratios, Section 3.5 (1995)
13. Federal Highway Administration: Manual Uniform Traffic Control Devices (November 2009), <http://mutcd.fhwa.dot.gov/>
14. American Association of State Highway and Transportation Official (AASHTO): Recommendations for AASHTO Superelevation Design (2003), <http://downloads.transportation.org/SuperelevationDiscussion9-03.pdf>
15. Cheng, Y., Lee, S.: A new method for quadratic curve detection using K-RANSAC with acceleration techniques. *Pattern Recognition* 28(5), 663–682 (1995)
16. Borkar, A., Hayes, M., Smith, M.T.: An efficient method to generate ground truth for evaluating lane detection systems. In: International Conference on Acoustics, Speech, and Signal Processing (2010)

An Analysis of the Road Signs Classification Based on the Higher-Order Singular Value Decomposition of the Deformable Pattern Tensors

Bogusław Cyganek

AGH University of Science and Technology
Al. Mickiewicza 30, 30-059 Kraków, Poland
cyganek@agh.edu.pl

Abstract. The paper presents a framework for classification of rigid objects in digital images. It consists of a generator of the geometrically deformed prototypes and an ensemble of classifiers. The role of the former is to provide a sufficient training set for subsequent classification of deformed objects in real conditions. This is especially important in cases of a limited number of available prototype exemplars. Classification is based on the Higher-Order Singular Value Decomposition of tensors composed from the sets of deformed prototypes. Construction of such deformable tensors is flexible and can be done independently for each object. They can be obtained either from a single prototype, which is then affinely deformed, or from many real exemplars, if available. The method was tested in the task of recognition of the prohibition road signs. Experiments with real traffic scenes show that the method is characteristic of high speed and accuracy for objects seen under different viewpoints. Implementation issues of tensor decompositions are also discussed.

1 Introduction

Recognition of objects in digital images is a key task of Computer Vision. However, the problem is complicated due to a great diversity of objects of interest, on the one hand, and limited information provided in digital images, on the other. Nevertheless, due to development of new classification methods and computational techniques, it is possible to construct some frameworks for fast and reliable recognition of at least some groups of well defined objects. In this paper we present one of such software frameworks. It can classify rigid objects detected in images which views are subject to a subgroup of projective transformations and noise. These unavoidable distortions are due to the geometrical and physical properties of the observed objects and conditions of image acquisition. The presented system relies mostly on the set of classifiers which perform a multilinear analysis of tensors which are composed of the prototype exemplars of objects. However, frequently the latter are not available in a sufficient number to allow recognition of geometrically transformed views of objects. Therefore to remedy this constraint the second important module of our framework is a generator of affinely deformed and noise conditioned artificial prototypes. Thus, the system can still reliably recognize an object even if only its single prototype is provided. Obviously, the more input prototypes, the better results can be obtained. The method is able to cope with different number of these for each object it is trained to recognize.

Application of tensors opened new possibilities for more precise analysis of complex data which depend on many different factors. Each such factor is represented by a new dimension of the tensor space (a mode of a tensor). In image processing different factors correspond to different viewpoints, illumination conditions, or geometric deformations of represented objects. This constitutes a qualitative difference compared to the matrix approach in which images characteristic of different viewing conditions had to be vectorized prior to the analysis, such as PCA [18]. Such tensor based methods have been already used for CV tasks as handwritten character classification [19] or face recognition [20], etc.

Specifically, in this paper we address the problem of reliable classification of the road signs (RS) based on their monochrome pictograms. In the aforementioned multilinear recognition framework, the task of RS classification is done with help of the Higher-Order Singular Value Decomposition (HOSVD, called also the Tucker decomposition) of the tensors built from the deformable versions of the prototype patterns of each of the pictograms. To the best of our knowledge, this is the first application of the HOSVD to the RS classification task. Nevertheless, as alluded to previously, the presented framework can be also used for recognition of another group of rigid objects, such as moving cars or fruits on a production line.

The work builds into our framework of RS recognition in which different detection and classification modules were reported in [6-9]. In the group of developed classifiers the presented in this paper tensor based method allows the best accuracy at very high speed of response and manageable occupation of memory. More information pertinent to the RS recognition task can be found in the works of de Escalera *et al.* [12], Paclik *et al.* [17], Chen *et al.* [4], or Bascón *et al.* [3], etc., as well as in the mentioned references [6-9].

The rest of the paper is organized as follows. We start with a discussion of the architecture of the system. Then details of the tensor representation of patterns and classification with the HOSVD are discussed. Further we discuss the implementation issues related to the object-oriented computer representation of tensors as well as to their so called proxy objects which allow efficient index manipulations of tensors without data copying. Finally, we present the experimental results and conclusions.

2 Architecture of the Road Signs Recognition System

Architecture of our object recognition framework applied the task of RS recognition is depicted in Fig. 1. It was designed to fit into our software framework developed during the recent years [6-9]. However, in this paper we focus mostly on the HOSVD based classification applied to the prohibition signs.

The preprocessing starts with the detection module which accepts an input color image and returns rectangular outlines of the compact red objects, as described in [9]. Such rectangles are then cropped and then their color signal is converted into a monochrome version by taking only the blue channel. Such a strategy showed to provide the best contrast of the pictograms of the prohibition signs. Then, the detected rectangle is registered to the size expected by the classification module, described in the next section, as described in [9].

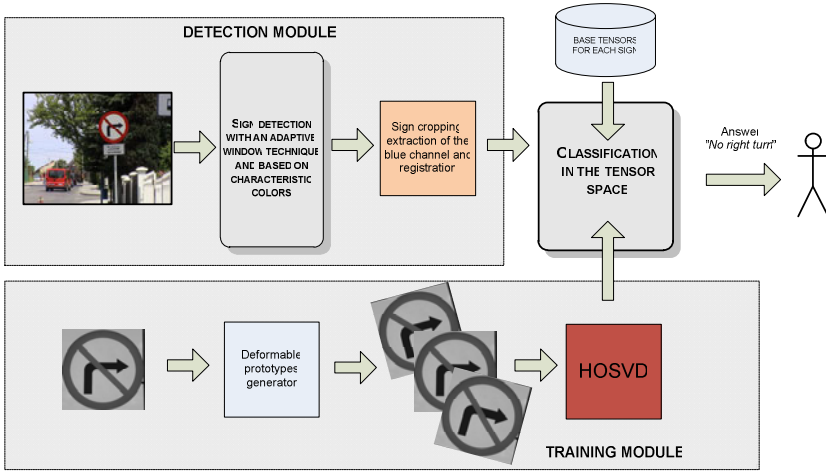


Fig. 1. Architecture of the road signs recognition system

However, to build the tensor space which is then HOSVD decomposed we use a set of prototypes extracted from real traffic scenes but the method works fine also for a set of the formal RS definitions (i.e. a law regulation). The patterns from the chosen set are then affinely transformed by the prototype exemplar generator. It was noticed that because the images are already registered to some common size by the detector, it is sufficient to constrain the affine transformations to pure rotations. Finally, the experiments showed that this method easily works with some small variations in horizontal/vertical positioning, i.e. by few pixels. Then, after HOSVD decomposition of such deformable patterns, only a number of dominating components is used to classify an incoming test pattern. The procedure is described in the next section.

3 Tensor Based Object Recognition

Tensors are mathematical objects used in many branches of science, such as mathematics and physics, due to their well defined transformation properties in respect to the change of a coordinate system [1]. However, in some applications, such as data mining, they are considered as multidimensional generalizations about matrices, i.e. the multidimensional arrays of data [5][16]. In this work we follow the second interpretation. Below, a short introduction to tensor decomposition is presented. More details can be found in references, e.g. [5][16][2].

Analogously to the matrix SVD decomposition [18], for a P dimensional tensor \mathcal{T} there exists a P -th order decomposition HOSVD. It allows each tensor $\mathcal{T} \in \mathfrak{R}^{N_1 \times N_2 \times \dots \times N_m \times \dots \times N_n \times \dots \times N_p}$ to be decomposed as follows

$$\mathcal{T} = \mathcal{Z} \times_1 \mathbf{S}_1 \times_2 \mathbf{S}_2 \dots \times_p \mathbf{S}_p, \tag{1}$$

where \mathbf{S}_k is a unitary mode matrix of dimensions $N_k \times N_k$, spanning the column space of the matrix $\mathbf{T}_{(k)}$ obtained from the mode- n flattening of \mathcal{T} ; $\mathcal{Z} \in \mathfrak{R}^{N_1 \times N_2 \times \dots \times N_m \times \dots \times N_n \times \dots \times N_p}$ is a core tensor of the same dimensions as \mathcal{T} , which satisfies the following conditions [16]:

1. Two subtensors $\mathcal{Z}_{n_k=a}$ and $\mathcal{Z}_{n_k=b}$, obtained by fixing the n_k index to a , or b , are orthogonal, i.e.

$$\mathcal{Z}_{n_k=a} \cdot \mathcal{Z}_{n_k=b} = 0, \tag{2}$$

for all possible values of k for which $a \neq b$.

2. All subtensors can be ordered according to their Frobenius norms

$$\|\mathcal{Z}_{n_k=1}\| \geq \|\mathcal{Z}_{n_k=2}\| \geq \dots \geq \|\mathcal{Z}_{n_k=N_p}\| \geq 0, \tag{3}$$

for all k .

The following Frobenius norm

$$\|\mathcal{Z}_{n_k=a}\| = \sigma_a^k \tag{4}$$

is called the a -mode singular value of \mathcal{T} . Each i -th vector of the matrix \mathbf{S}_k is the i -th k -mode singular vector.

Assuming decomposition (1) of a tensor \mathcal{T} , singular values (4) provide a notion of an energy of this tensor in the terms of the Frobenius norm, as follows

$$\|\mathcal{T}\|^2 = \sum_{a=1}^{R_1} (\sigma_a^1)^2 = \dots = \sum_{a=1}^{R_p} (\sigma_a^p)^2 = \|\mathcal{Z}\|^2, \tag{5}$$

where R_k denotes a k -mode rank of \mathcal{T} .

The SVD decomposition allows representation of a matrix as a sum of rank one matrices. The summation spans number of elements, however no more than a rank of the decomposed matrix. Similarly to the SVD decomposition of matrices, based on the decomposition (1), a tensor can be represented as the following sum

$$\mathcal{T} = \sum_{h=1}^{N_p} \mathcal{T}_h \times_p \mathbf{s}_p^h, \tag{6}$$

where

$$\mathcal{T}_h = \mathcal{Z} \times_1 \mathbf{S}_1 \times_2 \mathbf{S}_2 \dots \times_{p-1} \mathbf{S}_{p-1}, \tag{7}$$

denotes the basis tensors and \mathbf{s}_p^h are columns of the unitary matrix \mathbf{S}_p . Since \mathcal{T}_h is of dimension $P-1$ then \times_p in (6) is an outer product, i.e. a product of two tensors of dimensions $P-1$ and 1. The result is a tensor of dimension P , i.e. the same as of \mathcal{T} . Fig. 2 depicts a visualization of this decomposition for a 3D tensor. In this case \mathcal{T}_h

becomes two-dimensional, i.e. it is a matrix. Moreover, it is worth noting that due to orthogonality of the core tensor \mathcal{Z} in (7), \mathcal{T}_h are also orthogonal. Hence, \mathcal{T}_h in decomposition (6) constitute a basis. This is a very important result which allows construction of classifiers based on the HOSVD decomposition. Such a scheme is used in the proposed system for RS classification, although other tensor constructions with simultaneous data compression are also possible [19]. Nevertheless, in our case each set of prototypes for a *single* sign (i.e. a single class) is independently encoded as a separate tensor \mathcal{T} . This allows different numbers of prototypes in each of the classes. As alluded to previously, in each case the series (6) is usually truncated to the first $N \leq N_p$ most prominent components. In other words, a smaller but dominating N dimensional subspace is used to approximate \mathcal{T} .

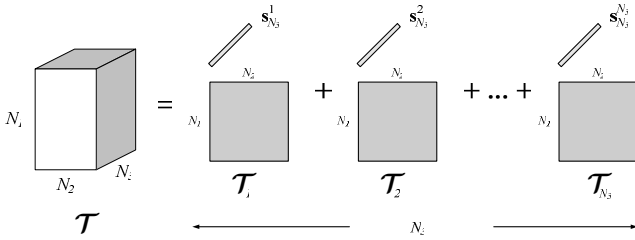


Fig. 2. Visualization of the tensor decomposition given by (6)

The series of *k-mode* products (7) can be equivalently represented in a matrix notation after tensor flattening

$$\mathbf{T}_{(k)} = \mathbf{S}_k \mathbf{Z}_{(k)} \left[\mathbf{S}_{k+1} \otimes \mathbf{S}_{k+2} \otimes \dots \otimes \mathbf{S}_p \otimes \mathbf{S}_1 \otimes \mathbf{S}_2 \otimes \dots \otimes \mathbf{S}_{k-1} \right]^T, \tag{8}$$

where \otimes denotes the Kronecker product. This provides us with a convenient link to the matrix representation of tensor equations which is discussed in the next section of this paper. By the same token, and taking into an account that \mathbf{S}_k are orthogonal, computation of the core tensor \mathcal{Z} can be expressed as

$$\mathbf{Z}_{(k)} = \mathbf{S}_k^T \mathbf{T}_{(k)} \left[\mathbf{S}_{k+1} \otimes \mathbf{S}_{k+2} \otimes \dots \otimes \mathbf{S}_p \otimes \mathbf{S}_1 \otimes \mathbf{S}_2 \otimes \dots \otimes \mathbf{S}_{k-1} \right]. \tag{9}$$

The HOSVD successively applies the matrix SVD decomposition to each of the flattened $\mathbf{T}_{(k)}$ versions of the input tensor \mathcal{T} . In result the \mathbf{S}_k matrices are computed [16]. In the 3D case and considering (9), the HOSVD can be written as

$$\mathbf{Z}_{(1)} = \mathbf{S}_1^T \mathbf{T}_{(1)} \left(\mathbf{S}_2 \otimes \mathbf{S}_3 \right). \tag{10}$$

As mentioned, in our framework the original tensor \mathcal{T}_i of a class i is obtained from the available exemplars of the prototype patterns for that class i . These, in turn, are obtained from the patterns cropped from the real traffic images which are additionally rotated in a given range (in our examples this was $\pm 12^\circ$ with a step of 2°) with

additionally added normal noise (a procedure for this is described in [10]). Since for different signs a different number of exemplars is available, such a strategy allows each pattern to be trained with different number of prototypes. Finally, the training stage ends in computation of \mathcal{T}_h^i for each \mathcal{T}_i , in accordance with (7).

Recognition is done by testing approximation of a given pattern P_x in each of the spaces spanned by the set of bases \mathcal{T}_h given in (6). This is done by solving the following minimization problem

$$\min_{c_h^i} \left\| P_x - \sum_{i=1}^N c_h^i \mathcal{T}_h^i \right\|^2, \quad (11)$$

where c_h^i are the coordinates of P_x in the manifold spanned by \mathcal{T}_h^i . Due to the orthogonality of the tensors \mathcal{T}_h^i , the above reduces to the maximization of the following parameter [19]

$$\rho_i = \sum_{i=1}^N \left\langle \hat{\mathcal{T}}_h^i, \hat{P}_x \right\rangle^2, \quad (12)$$

where the $\langle \cdot, \cdot \rangle$ operator denotes the scalar product of the tensors. The returned by a classifier pattern is a class i for which the corresponding ρ_i from (12) is the largest. In our system we set a threshold ($\tau=0.85$); Below this threshold the system answers “*don't know*”. Such a situation arises if wrong pattern is provided by the detector or a sign which system was not trained for. The number N in (12) of components was set from 3 to 9. The higher N , the better fit, though at an expense of computation time.

4 Computer Representation of the Flat Tensors

Many platforms have been developed for efficient tensor representations. However, sometimes they lack sufficient elasticity of using different data types or they do not fit into the programming platforms [2][5]. In this paper we address the problem of efficient tensor representation and manipulation in software implementations. Our main assumptions can be summarized as follows.

1. Flexibility in accessing tensors as multidimensional arrays and flat data representations at the same time without additional copies.
2. Efficient software and/or hardware processing.
3. Flexible element type selection and specializations for tensors.

A proposed class hierarchy for storage and manipulation of tensors is shown in Fig. 3.

The base template class *TImageFor*<> comes from the HIL library [14]. The library is optimized for image processing and computer vision tasks, as well as for fast matrix operations [10]. *TFlatTensorFor*<> is the base class for tensor representation. Thus, in our framework a tensor is represented as a specialized version of a matrix class. This does not follow the usual way in which a matrix is seen as a special two-dimensional tensor. This follows from the fact that tensors in our system are always stored in the flattened representation for a given mode. This also follows a

linear organization of the computer memory. In these terms a tensor is just a data structure with a tensor mode governed by the more specialized objects in the hierarchy. At the same time, flexible and efficient methods developed for manipulation of the matrix data objects [14] are retained. In other words, the “is a” relationship showed up to be a more practical solution than the previously tried “has a”, in which a tensor object contained a matrix that stored its data in one mode. In effect the *TFlatTensorFor*<> has two sets of the principal methods for accessing its elements. The first pair *Get/SetElement* takes as an argument a *TensorIndex* which is a vector of indices. Its length equals dimension of the tensor. The second pair of functions *Get/SetPixel* is inherited from the base *TImageFor*<>. The latter allow access to the matrix data providing simply its row and column (*r,c*) indices.

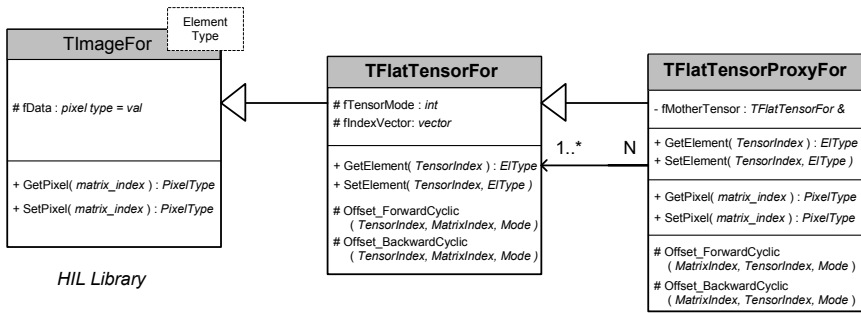


Fig. 3. A tensor class hierarchy

The *TFlatTensorProxyFor*<> class is a simplified proxy pattern to the *TFlatTensorFor*<> [10]. These are useful in all cases in which tensor representations in different flat *n*-modes are necessary. Proxies allow this without creating a copy of the input tensor which could easily consume large parts of memory and time. An example is the already discussed HOSVD decomposition. In each step of this algorithm the *n*-mode flat tensor needs to be created from the initial tensor **T**, for all *n*'s [16]. In our realization these two-way index transformations are possible with the *Offset_ForwardCyclic/Offset_BackwardCyclic* methods which recompute tensor-matrix indices in two ways and in two cyclic modes (backward and forward), and also for different *n*-modes. More specifically, an index of an element in a tensor **T** of dimension *k* is given by a tuple (*i*₁, *i*₂, ..., *i*_{*k*}) of *k* indices. This maps into an offset *q* of a linear memory

$$q = (((i_1 n_2 + i_2) n_3) + \dots) n_k + i_k, \tag{13}$$

where the tuple (*n*₁, *n*₂, ..., *n*_{*k*}) gives dimensions of **T**. On the other hand, matrix representation always involves selection of two dimensions $(r,c) = \left(n_m, \prod_{z=1, z \neq m}^k n_z \right)$, *m* equals a mode of the **T**. In consequence, an element at index *q* has to fit into such a matrix. In the tensor proxy pattern the problem is inverted - given a matrix offset *q* a corresponding tensor index tuple has to be determined due to different modes of the

tensors. This is obtained by successive division of the q in (13) by n_p for starting from $p=k$ up to $p=1$, since for all k it holds that $i_k < n_k$. A series of indices i_p is obtained in a form of residua of such successive divisions. Summarizing, the advantages of the proposed tensor classes are as follows:

1. A uniform treatment of the tensor as well as its matrix n -modes. A tensor is stored only in a single chosen mode while other modes are obtained exclusively by index manipulations.
 2. Tensor proxy objects allow simultaneous manipulation of a tensor in its all possible n -mode flat representations without data copying.
 3. Template implementation allows different types of tensor elements (such as float, boolean or fixed-point formats).
 4. Object oriented C++ implementation can be easily ported to other OO languages such as C#, Java, Python, etc.
- The described tensor software framework can be accessed from the Internet [11].

5 Experimental Results

The presented object classification framework was implemented in C++. Experiments were run on a computer with 2GB RAM and Pentium Core 2 T7600 @ 2.33GHz.



Fig. 4. Exemplary real traffic scenes used in the experiments. The method is able to correctly recognize signs of different size and orientation.

Fig. 4 and Fig. 5 depict two real traffic scenes with signs correctly detected and then classified by the presented HOSVD based system. Despite inherent rotation, as well as variations of tint and different lighting, the signs were recognized correctly.

Fig. 6 Fig. 7 depict the first five tensors \mathcal{T}_h and the corresponding five core tensors \mathcal{Z}_n which were computed for the for the “40 km/h speed limit” and “No pass” signs, respectively. An inherent rotation added during training is well visible.

An average accuracy of recognition was measured in terms of the error rate which plot depicts Fig. 8b. However, during the tests it was observed that some signs cause more errors (such as e.g. the STOP sign), whereas the other can be recognized very reliably. This is caused mostly by specific pictogram distribution of

different signs. Some signs are also very similar, especially if geometrically changed, e.g. 30 km/h compared to the 80 km/h, etc.

As alluded to previously, the method tolerates well imperfections in detection (such as not well aligned window, etc.), as well as variations in color tint and/or lighting obtained in real road conditions (we used the Marlin® and Sony® cameras). The method is also resistant to the slight projective deformations (i.e. allowed for the



Fig. 5. Examples of correct classification of the signs despite the imprecise detection and under different color and/or lighting conditions. In all cases the color images were converted to the monochrome versions by taking exclusively the blue channel. Then intensity values were conditioned by histogram equalization method.

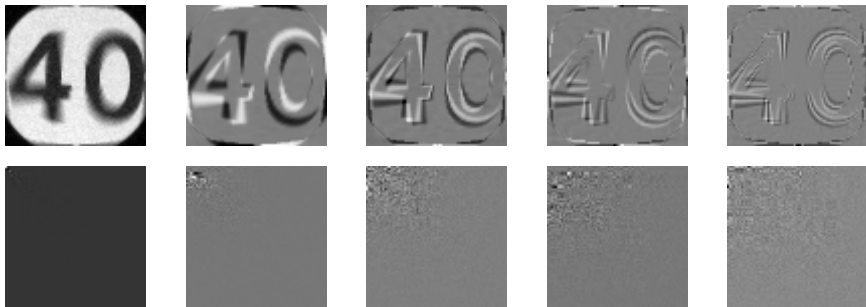


Fig. 6. First five tensors \mathcal{T}_h for the 40 km/h speed limit sign (upper row), and the corresponding five core tensors \mathcal{Z}_n (lower row)

road signs in respect to the drivers' direction of view), as well as to some occlusions if these do not affect the main part of the pictogram. Correct operations under different operating conditions are presented in Fig. 4 and Fig. 5, for instance.

Training of the data base in Fig. 8a takes around 8-9s in our platform, while run time classification is in order of 0.04-0.07s per single image of resolution 640×480, depending on a size of the test pattern (the difference in computation time depends on time necessary for the geometrical registration to the test pattern).

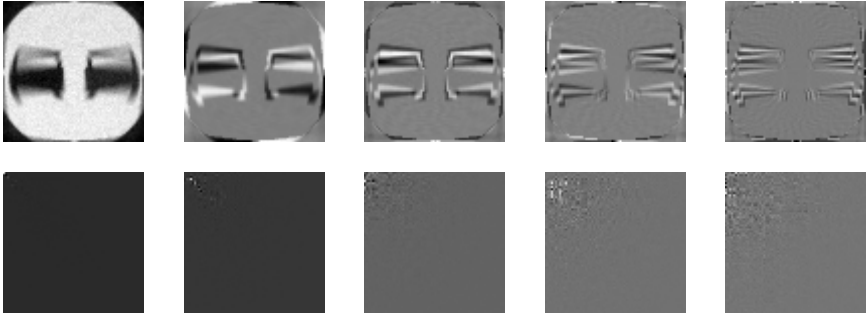


Fig. 7. First five tensors \mathcal{T}_h for the “No pass” sign (upper row), and the corresponding five core tensors \mathcal{Z}_n (lower row)

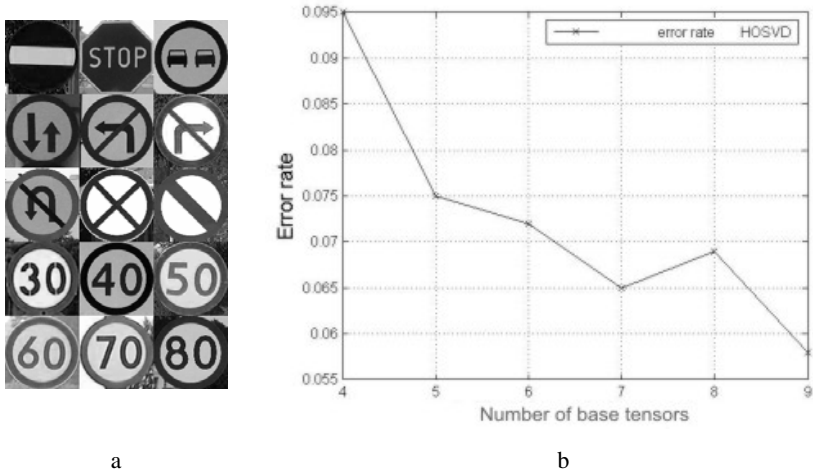


Fig. 8. The data-base of prototype exemplars from which the rotated and noisy patterns were constructed from which the tensors are composed (a). Accuracy rate of the pictogram classification system in respect to the number of N base tensors in (12) (b).

6 Conclusions

In this paper a software framework for rigid object classification is presented which was applied to the task of road signs recognition. The classification method relies on the Higher-Order Singular Value Decomposition of the deformable prototype tensors. These, in turn, are built for each pictogram from deformable versions of its real prototypes. The group of deformations contains only rotations since small shifts are well compensated without explicit training. Additionally, this way prepared prototypes were endowed with the Gaussian noise to enhance robustness. The method was tested with images containing real traffic scenes. Compared to other solutions it can be characterized as showing the highest accuracy and recognition speed. The method is also resistant to the small projective deformations of the observed signs, as

well as to the slight variations in color, lighting conditions and occlusions which do not obscure the pictogram. The most troublesome are situations in which an object provided by the detector does not belong to any of the patterns used during training. To cope with such outliers a match threshold was set based on experiments. The obtained accuracy on the group of prohibition signs reached 95% at speed of 15-25 frames/s of resolution 640×480. Additionally, we provide software for efficient representation and manipulations of tensors, as well as for their decomposition.

Acknowledgement

This work was supported from the Polish funds for scientific research in 2010.

References

1. Aja-Fernández, S., de Luis García, R., Tao, D., Li, X. (eds.): *Tensors in Image Processing and Computer Vision*. Springer, Heidelberg (2009)
2. Bader, B.W., Kolda, T.G.: *MATLAB Tensor Classes for Fast Algorithm Prototyping*. *ACM Transactions on Mathematical Software* 32(4), 635–653 (2006)
3. Bascón, S.M., Rodríguez, J.A., Arroyo, S.L., Caballero, A.F., López-Ferreras, F.: An optimization on pictogram identification for the road-sign recognition task using SVMs. *Computer Vision and Image Understanding* 114(3), 373–383 (2010)
4. Chen, X., Yang, J., Zhang, J., Waibel, A.: Automatic Detection and Recognition of Signs from Natural Scenes. *IEEE Trans. on Image Proc.* 13(1), 87–99 (2004)
5. Cichocki, A., Zdunek, R., Phan, A.H., Amari, S.-I.: *Nonnegative Matrix and Tensor Factorizations*. Wiley, Chichester (2009)
6. Cyganek, B.: Soft System for Road Sign Detection. In: *Theory and Applications of Fuzzy Logic and Soft Computing, Advances in Soft Computing*, vol. 41, pp. 316–326. Springer, Heidelberg (2007)
7. Cyganek, B.: Real-Time Detection of the Triangular and Rectangular Shape Road Signs. In: Blanc-Talon, J., Philips, W., Popescu, D., Scheunders, P. (eds.) *ACIVS 2007. LNCS*, vol. 4678, pp. 744–755. Springer, Heidelberg (2007)
8. Cyganek, B.: Color Image Segmentation with Support Vector Machines: Applications To Road Signs Detection. *International Journal of Neural Systems* 18(4), 339–345 (2008)
9. Cyganek, B.: Circular Road Signs Recognition with Soft Classifiers. *Integrated Computer-Aided Engineering* 14(4), 323–343 (2007)
10. Cyganek, B., Siebert, J.P.: *An Introduction to 3D Computer Vision Techniques and Algorithms*. Wiley, Chichester (2009)
11. Cyganek, B.: FlatTensor hierarchy (2010), <http://home.agh.edu.pl/~cyganek/FlatTensor.zip>
12. de la Escalera, A., Armingol, J.A.: Visual Sign Information Extraction and Identification by Deformable Models. *IEEE Transactions On Intelligent Transportation Systems* 5(2), 57–68 (2004)
13. Duda, R., Hart, R., Stork, D.: *Pattern Classification*. Wiley, Chichester (2001)
14. http://www.wiley.com/legacy/wileychi/cyganek3dcomputer/supp/HIL_Manual_01.pdf
15. Kuncheva, L.: *Combining Pattern Classifiers. Methods and Algorithms*. Wiley, Chichester (2004)

16. de Lathauwer, L., de Moo, B., Vandewalle, J.: A Multilinear Singular Value Decomposition. *SIAM Journal Matrix Analysis and Applications* 21(4), 1253–1278 (2000)
17. Paclík, P., Novovičová, J., Duin, R.P.W.: Building road sign classifiers using a trainable similarity measure. *IEEE Transactions on Intelligent Transportation Systems* 7(3), 309–321 (2006)
18. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes in C. In: *The Art of Scientific Computing*, 2nd edn., Cambridge University Press, Cambridge (1999)
19. Savas, B., Eldén, L.: Handwritten digit classification using higher order singular value decomposition. *Pattern Recognition* 40, 993–1003 (2007)
20. Vasilescu, M., Terzopoulos, D.: Multilinear analysis of image ensembles: Tensorfaces. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) *ECCV 2002*. LNCS, vol. 2350, pp. 447–460. Springer, Heidelberg (2002)

An Effective Rigidity Constraint for Improving RANSAC in Homography Estimation

David Monnin, Etienne Bieber, Gwenaël Schmitt, and Armin Schneider

French-German Research Institute of Saint-Louis (ISL), 5 rue du Général Cassagnou,
PO Box 70034, 68300 Saint-Louis, France
david.monnin@isl.eu

Abstract. A homography is a projective transformation which can relate two images of the same planar surface taken from two different points of view. Hence, it can be used for registering images of scenes that can be assimilated to planes. For this purpose a homography is usually estimated by solving a system of equations involving several couples of points detected at different coordinates in two different images, but located at the same position in the real world. A usual and efficient way of obtaining a set of good point correspondences is to start from a putative set obtained somehow and to sort out the good correspondences (inliers) from the wrong ones (outliers) by using the so-called RANSAC algorithm. This algorithm relies on a statistical approach which necessitates estimating iteratively many homographies from randomly chosen sets of four-correspondences. Unfortunately, homographies obtained in this way do not necessarily reflect a rigid transformation. Depending on the number of outliers, evaluating such degenerated cases in RANSAC drastically slows down the process and can even lead to wrong solutions. In this paper we present the expression of a lightweight rigidity constraint and show that it speeds up the RANSAC process and prevents degenerated homographies.

1 Introduction

In the field of computer vision, homographies are widely used to relate images of scenes assimilable to planar surfaces. All typical homography applications from the computation of camera motion to image mosaicing, video stabilization, augmented reality, image rectification or sub-pixel resolution extrapolation rely in a way on image registration. Homographies are consequently expected to reflect a mapping from one image plane to another which corresponds to a rigid-body transformation. It is therefore assumed that a rigid body keeps its shape during the acquisition of images to be related and that only its projection on the image plane changes when the camera view changes.

The homogeneous coordinates representation used in projective geometry, which is briefly described hereafter, allows a very synthetic and convenient matrix representation of a homography. Unlike in Euclidean geometry, a combination of 3D rotation and translation necessitates only one matrix multiplication.

Thanks to this mathematical representation, it is very easy, from a mathematical point of view, to estimate the homography parameters from a set of point correspondences taken from two different images representing the same real-world location at different image coordinates. Unfortunately, even the best feature space based methods have performance limitations and in practical cases it is not always trivial to detect and perfectly associate points from two images that correspond to the same real-world location. It is therefore common practice to use the RANSAC algorithm [1] to sort out putative point correspondences obtained by the mean of some feature space methods. This algorithm delivers both an estimation of the homography and a set of point correspondences which are consistent with this estimate. Even if RANSAC is known to be very robust, it can possibly fail and lead to results which do not reflect a rigid-body transformation. This was the motivation for the investigation reported in this paper, which led us to clarify the fact that a homography cannot be reduced to a rigid-body transformation and that RANSAC is not always able to reject non-rigid-body transformations. Finally, we present a lightweight rigidity constraint that not only allows RANSAC to avoid some degenerated homographies, but also speeds up the whole process in unexpected proportions.

2 Backgrounds of Homography Estimation

2.1 Homography Estimation and Image Registration

A *homography* is a *projective transformation* also called *projectivity* or *collineation* defined by an invertible mapping h from the projective plane \mathbb{P}^2 to itself that maps lines to lines [2,3]. Points in \mathbb{P}^2 are described by column 3-vectors of the form $p = (x_1, x_2, x_3)^\top$ defining their so-called *homogeneous coordinates*. In homogeneous coordinates, given a non-zero constant k , the set of vectors $(k \cdot x_1, k \cdot x_2, k \cdot x_3)^\top$ describes the same point of \mathbb{P}^2 . A representation of an arbitrary point $(x_1, x_2, x_3)^\top$ from \mathbb{P}^2 in the Euclidean plane defined in \mathbb{R}^2 can be obtained by the usual normalization $(x_1/x_3, x_2/x_3, 1)^\top$ of the homogeneous coordinates which leads to the Euclidean coordinates $(x, y)^\top = (x_1/x_3, x_2/x_3)^\top$. In the same way, a point from the Euclidean plane defined by a column 2-vector $(x, y)^\top$ in \mathbb{R}^2 can be represented in \mathbb{P}^2 by the 3-vector $(x, y, 1)^\top$. It is important to remember that $(x, y, 1)^\top$ is not the unique representation of $(x, y)^\top$ in \mathbb{P}^2 as it is by definition equivalent to the set of 3-vectors $(k \cdot x, k \cdot y, k)^\top$. Hence, given a 3×3 homography matrix H and two points p and p' , the projective transformation which maps p to p' is written:

$$p' = H \cdot p. \quad (1)$$

Using the homogeneous coordinates of p and p' this projective transformation can be expressed in the matrix form as:

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \quad (2)$$

or alternatively in the form of an equivalent system of equations:

$$\begin{cases} x'_1 = h_1 \cdot x_1 + h_2 \cdot x_2 + h_3 \cdot x_3 \\ x'_2 = h_4 \cdot x_1 + h_5 \cdot x_2 + h_6 \cdot x_3. \\ x'_3 = h_7 \cdot x_1 + h_8 \cdot x_2 + h_9 \cdot x_3 \end{cases} \quad (3)$$

Scaling H with a non-zero scalar k yields $p' = (k \cdot H) \cdot p$ or $p' = H \cdot (k \cdot p)$ according to the commutativity property, which is by definition equivalent to (1) since $k \cdot p = p$ in homogeneous coordinates. Matrix H is thus said to be homogeneous, since, similarly to the homogeneous representation of a point, only the ratios of its elements are significant. Given that with nine parameters there are eight possible ratios of parameters, a homography has eight degrees of freedom. Without loss of generality, when estimating the parameters of a homography, it is then convenient for the uniqueness of the representation to use a normalized representation of H such that $h_9 = 1$.

In image registration tasks, p and p' represent pixels in two different images and are therefore more naturally described by their respective Euclidean coordinates $(x, y)^\top$ and $(x', y')^\top$. Using the corresponding homogeneous representation $(x, y, 1)^\top$ and $(x', y', 1)^\top$ of these Euclidean coordinates, the projective transformation expressed in (2) becomes:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \quad (4)$$

where H is in a normalized form. Given that in homogeneous coordinates $p = (x, y, 1)^\top = (x_1/x_3, x_2/x_3, 1)^\top$ and $p' = (x', y', 1)^\top = (x'_1/x'_3, x'_2/x'_3, 1)^\top$, we have $x = x_1/x_3$, $y = x_2/x_3$, $x' = x'_1/x'_3$ and $y' = x'_2/x'_3$, which using values defined in (3), leads to:

$$x' = \frac{h_1 \cdot x + h_2 \cdot y + h_3}{h_7 \cdot x + h_8 \cdot y + 1} \quad (5)$$

and

$$y' = \frac{h_4 \cdot x + h_5 \cdot y + h_6}{h_7 \cdot x + h_8 \cdot y + 1}. \quad (6)$$

Hence one couple of points leads to two equations. With eight degrees of freedom, at least four couples of points leading to eight equations are then necessary to estimate all homography parameters. Methods of detecting and associating points from two different images are beyond the scope of this paper, but it is well-known that point correspondences are rarely perfectly reliable until they are sorted out by means of the RANSAC algorithm. RANSAC is particularly useful when the number of bad point correspondences, called *outliers*, is large with regard to the number of good point correspondences, called *inliers*.

Algorithm 1. RANSAC algorithm for homography estimation

```

number_of_iterations := 0;
inliers := {};
H := {};

repeat
  random_sample := four randomly-selected correspondences;

  if (is_not_degenerated(random_sample))
  begin
    current_H := homography_processed_from(random_sample);
    current_inliers := putative correspondences matching H;
    if (number_of(current_inliers) > number_of(inliers))
    begin
      inliers := current_inliers;
      H := current_H;
    end
  end

  number_of_iterations := number_of_iterations + 1;
until (number_of_iterations > max_number_of_iterations)

return (inliers,H);

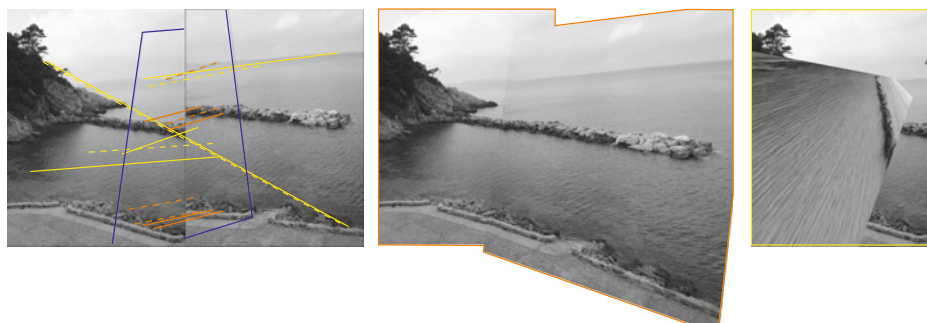
```

2.2 RANSAC in Homography Estimation

The *Random Sample Consensus* algorithm, or RANSAC, is an iterative method of estimating the parameters of a mathematical model from sample data containing both inliers and outliers, with the ability to simultaneously sort out the inliers from the outliers according to the estimated model [1]. This algorithm, described in Algorithm 1, is commonly used for homography estimation in image registration tasks. For this purpose, it starts with a putative set of point correspondences from two different images. Samples of four point correspondences are then iteratively evaluated by first processing a homography using the four correspondences and then by checking the consistency of all the putative correspondences with respect to this homography. The consistency of a correspondence with a given homography can be evaluated using different error measurement methods [3] such as, for example, the symmetric transfer function:

$$\epsilon = d(p, H^{-1} \cdot p')^2 + d(p', H \cdot p)^2. \quad (7)$$

The process ends after a number of iterations which is interactively re-evaluated with respect to the largest current number of inliers [3]. The literature recommends that degenerated samples containing three collinear points should not be evaluated, as it leads to under-determined systems of equations [2,3]. Another current advice is to prefer samples with a good spatial distribution over the images. If the first advice definitely makes sense, the second one is more difficult to follow in the case of images with very little overlapping where the inliers are concentrated in regions much smaller than the size of images. Even



a) Two sets of correspondences sorted out using RANSAC (plain lines: best random sample, dashed lines: other correspondences) b) Correct registration obtained from the orange set of correspondences c) Wrong registration obtained from the yellow set of correspondences

Fig. 1. Two different results of RANSAC in a difficult image registration task

when RANSAC — yet known to be very robust — is used, dealing with images with little overlapping leads to difficulties in some image registration tasks. The kind of issues encountered is illustrated in figure 1 where the image registration was done following an approach similar to the one in [415]. In this example, the overlap is of about thirty per cent, which is usually enough for a fairly good registration. In this case, however, the situation is more difficult, as many point correspondences are detected in the sea area which is different in both images. This introduces a lot of outliers and reduces the number of possible inliers in the overlapping region, so that the total amount of inliers is finally estimated at eight only. In the best case, it is possible to get the result of figure 1b). However, depending on the random sample selection, the result of figure 1c) was also obtained. This less glorious result suggests that some kinds of degenerated samples led to an absurd homography for which eight bad point correspondences were unfortunately consistent. Anyway, in this case the wrong solutions compete with the good one and the result is somewhat uncertain, which justifies a further investigation.

3 A Rigidity Constraint for Improving RANSAC

3.1 Analysis of the Invalid Homographies Obtained with RANSAC

When analyzing the cases where RANSAC fails to give a good result, it appears that most often, the resulting homographies do not reflect a rigid-body transformation. Figure 2 illustrates quite well the kind of homographies which can occur in difficult situations where the number of outliers is high with respect to the number of inliers. The homographies of figure 2 were obtained by artificially imposing a geometrical rectification on the original image of figure 2a) in a way which has nothing in common with a rigid-body transformation. In these examples, the top-left, top-right, bottom-left and bottom-right corners were shifted

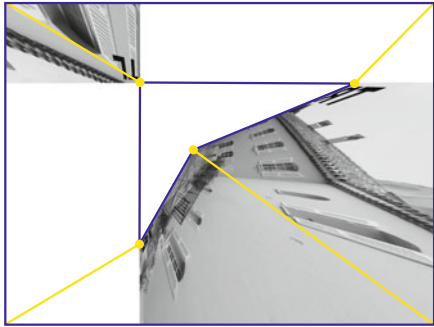


a) original image (1600 × 1200 pixels)



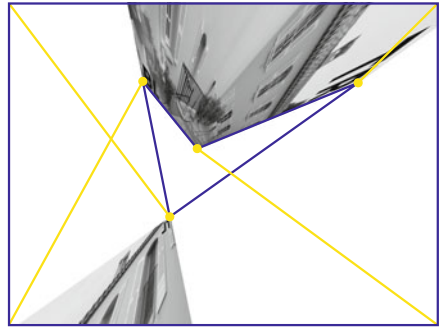
b) projective transformation using

$$H = \begin{pmatrix} -1.048 & 0.000 & 523.8 \\ -0.857 & 0.571 & 257.1 \\ -0.001 & 0.000 & 1 \end{pmatrix}$$



c) projective transformation using

$$H = \begin{pmatrix} -0.794 & 0.000 & 397.0 \\ 0.000 & -0.476 & 142.9 \\ -0.001 & -0.001 & 1 \end{pmatrix}$$



d) projective transformation using

$$H = \begin{pmatrix} 2.072 & -0.414 & -911.9 \\ -0.933 & -1.306 & 1604 \\ 0.001 & -0.002 & 1 \end{pmatrix}$$

Fig. 2. A set of homographies which does not reflect a rigid-body transformation

toward the center of the image in different ways. The linear displacements are represented by yellow lines in figure 2. Considering the initial rectangular shape formed by the corners of the original image and those resulting from the shifted corners, it is clear that the geometrical transformations imposed here cannot be obtained by any rigid-body transformation. In figure 2b) the order of the corners is changed by inverting the top-right and bottom-right corners, thus leading to a kind of bow-tie shape. In figure 2c) and 2d), the convexity of the initial shape is modified and in figure 2d) the order of the top-left and bottom-left corners is also inverted. However, even if it does not reflect a rigid-body transformation, a homography exists for any set of four correspondences. The beginnings of an explanation for this matter lie in the fact that the projective plane \mathbb{P}^2 is not the Euclidean plane of images and has a very different topology [3]. Even if we start to work with Euclidean coordinates, the projective transformations

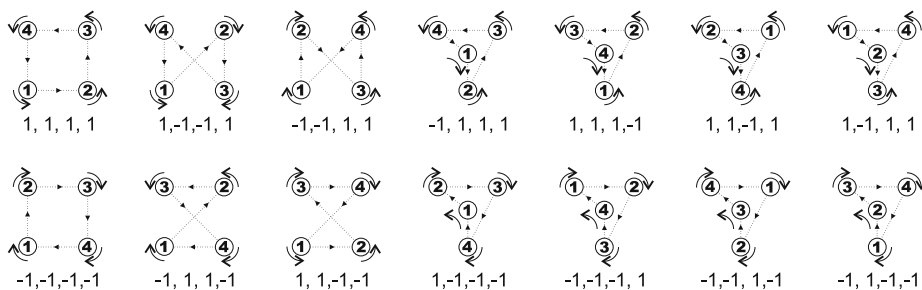


Fig. 3. All possible rotation-invariant oriented closed paths passing through four points

are evaluated using homogeneous coordinates in the projective plane \mathbb{P}^2 and the solution is then projected back into the Euclidean plane of an image. But as the projective plane \mathbb{P}^2 is two-sheeted and not simply connected, the result can be very surprising once projected into the simply-connected Euclidean plane.

Given that a homography exists for any four-correspondences set, it is impossible for RANSAC to reject degenerated homographies when the number of correspondences consistent with them equals or is greater than the number of inliers. A solution could come out from the ability to decide, from each sample of correspondences to be evaluated by RANSAC, whether a particular sample can possibly lead to a rigid-body transformation or not.

3.2 Toward a Rigidity Constraint for Improving RANSAC

In the previous analysis of some invalid homographies, it was shown that when considering the shapes formed by the four point correspondences, before and after the projective transformation, modifying the relative order of the corners of a shape or changing its convexity leads to a non-rigid-body transformation. In other words, for a homography to correspond to a rigid-body transformation, regardless of any rotation or relative distance variation between the corners, a fully convex shape has to be related to a fully convex shape and a shape with one concavity has to be related to another shape with one concavity. Additionally, in the case of shapes with one concavity, the point correspondences have to be correctly ordered for the concavities to be related by the same correspondence. In a first approach, the basic idea of checking for a rigid-body transformation from a randomly chosen set of four-correspondences, could be to order the point correspondences and link them together to form either a fully concave shape or a shape with one concavity, given that with four points, it can only be one or the other. Then, it has to be verified whether the shape correspondence preserves the convexity, and in the case of a concavity in the shapes, whether the concavity is related by the same correspondence. But this approach would be by far too complex. A better solution consists of keeping the initial random order of the point correspondences and of finding a criterion which makes it possible to confirm whether the shape transformation is consistent with a rigid-body transformation or not. In order to do so, it is possible to consider, for both the

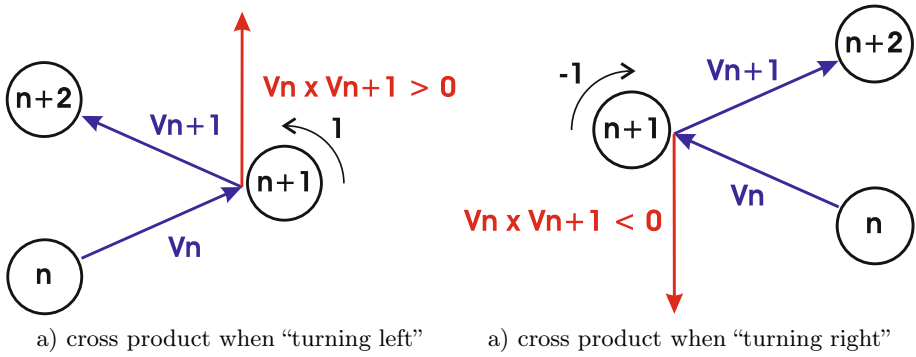


Fig. 4. Sign of the cross product as a function of the kind of “turn” in an oriented path

fully convex shapes and the shapes with one concavity, any possible closed path linking their four points and oriented in such a way as to reflect the order of the points. There are exactly fourteen possibilities represented in figure 3. Let us now consider the different direction turns necessary to follow the paths and code each path in a sequence of “1” and “-1” corresponding to “turn left” and “turn right”, respectively. It is obvious, from the four-digit chain codes reported in figure 3, that the codes formed in this way are unique and that each of them represents a different path. In the context of image registration, it means that if the chain code formed from the four points of a first image is the same as the one obtained from the corresponding points in a second image, the shapes they constitute are the same in both images and the point correspondences are ordered in the same way. Hence, this condition is sufficient and necessary for the homography estimation based on these four point correspondences to produce a rigid-body transformation.

3.3 Mathematical Expression of the Rigidity Constraint

The expression of the chain code introduced in the previous section does not necessitate processing the value of the angle formed by the three consecutive points of an oriented path. Indeed, as it is only important to identify the direction of the turns, let us simply consider the cross product $V_n \times V_{n+1}$ of the vectors V_n and V_{n+1} which are respectively defined from point n to point $n + 1$ and from point $n + 1$ to point $n + 2$. Figure 4 highlights that the direction turns are simply given by the direction, i.e. the sign, of the cross product $V_n \times V_{n+1}$. The rigidity constraint can then be expressed as:

$$sign(V_n \times V_{n+1}) = sign(V'_n \times V'_{n+1}) \quad \forall n = 1..4, \text{ with } V_5 = V_1 \text{ and } V'_5 = V'_1, \tag{8}$$

where vectors V_n and V_{n+1} are formed from points of a first image, while V'_n and V'_{n+1} are formed from their correspondents in a second image. Given $(x_n, y_n, 0)^T$, the Euclidean coordinates of a pixel p_n , a vector V_n can be written:

$$V_n = (V_{x_n}, V_{y_n}, 0)^T = (x_{n+1} - x_n, y_{n+1} - y_n, 0)^T. \tag{9}$$

Then the definition of the cross product $V_n \times V_{n+1}$ can also be expressed as the determinant of the following matrix, where \vec{i} , \vec{j} , \vec{k} are the unit vectors of the standard basis:

$$V_n \times V_{n+1} = \det \begin{pmatrix} \vec{i} & \vec{j} & \vec{k} \\ V_{x_n} & V_{y_n} & 0 \\ V_{x_{n+1}} & V_{y_{n+1}} & 0 \end{pmatrix}, \quad (10)$$

which gives:

$$V_n \times V_{n+1} = (V_{x_n} \cdot V_{y_{n+1}} - V_{y_n} \cdot V_{x_{n+1}}) \cdot \vec{k}. \quad (11)$$

Based on this formulation, the final expression of the rigidity constraint [\(8\)](#) can be written as:

$$\begin{aligned} \text{sign}(V_{x_n} \cdot V_{y_{n+1}} - V_{y_n} \cdot V_{x_{n+1}}) &= \text{sign}(V'_{x_n} \cdot V'_{y_{n+1}} - V'_{y_n} \cdot V'_{x_{n+1}}) \quad \forall n = 1..4, \\ &\text{with } V'_5 = V_1 \text{ and } V'_5 = V'_1. \end{aligned} \quad (12)$$

This rigidity constraint has to be added to the `is_not_degenerated()` test function of Algorithm [1](#). It must be noticed that the collinearity test commonly suggested can also be performed by evaluating the cross product $V_n \times V_{n+1}$. If it equals zero, the three consecutive points are collinear and if it is very small, they are quasi-collinear. Thus, the rigidity constraint we are proposing does not require any additional processing cost.

Even if the rigidity constraint proposed here has been investigated following our own approach for the purpose of homography estimation, it can be related to an extension of the geometrical constraint proposed in [\[6\]](#) for affine homographies. Those simplified homographies have only six degrees of freedom and are obtained from only three point correspondences [\[3\]](#). Our approach should hopefully strengthen the assumption made in [\[6\]](#) concerning an extension of the geometrical constraint for affine homographies to the case of projective transformations.

Additionally, when the chain codes identified in the first row are compared to those immediately below them in the second row of figure [3](#), it clearly appears that they represent a mirrored version of the same oriented path and that they only differ by their signs. This makes it quite trivial to loosen the rigidity constraint in order to allow the registration of images even if a number of them are mirrored images. This can be useful in cases where it is necessary to register images from film negatives or slides, the correct side of which is unknown. The rigidity constraint has then to be adapted; it simply consists of changing the sign of the elements of one of the chain codes to be compared, if and only if, the first elements tested in the two chain codes differ.

4 Evaluation Tests

In the context of image registration, it is clear that if point correspondences are perfectly chosen and therefore are all inliers, the resulting homography will be a rigid-body transformation. Besides, if all the point correspondences are inliers

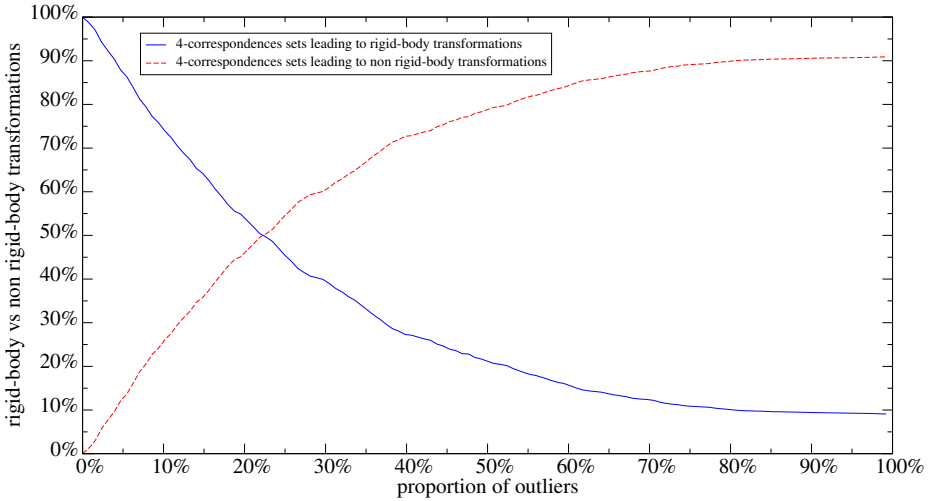


Fig. 5. Proportion of four-correspondence sets evaluated by RANSAC leading to either a rigid-body or a non-rigid-body transformation with respect to the proportion of outliers

leading to a rigid-body transformation, so is any subset of four correspondences. Hence, non rigid-body transformations only occur in homography estimation from four correspondences, when outliers are introduced. In order to evaluate the proportion of rigid-body and non-rigid-body transformations obtained when estimating a homography from randomly chosen four-correspondences sets, a set of 256 inliers regularly distributed over an image was defined. Then the proportion of inliers and outliers was progressively modified by successively exchanging the corresponding points of two inliers, which increases the number of outliers by two and decreases the number of inliers by two, respectively. At each step, 10^7 randomly chosen four-correspondences sets were evaluated for rigidity and the results are illustrated in figure 5. It shows that with only 22% of outliers, half the homographies obtained from four correspondences are non-rigid-body transformations. It means that when RANSAC has to deal with 22% outliers or more, it spends more than half its processing time considering solutions that have no chance of success.

The computing cost for processing the rigidity constraint is nearly negligible compared to the one needed for estimating a homography and checking all the putative correspondences for consistency with this homography. Thus, when using the rigidity constraint, the overall RANSAC processing time is only devoted to rigid-body transformations. It is then possible to express the speed-up factor s obtained thanks to the rigidity constraint in the following way:

$$t \cdot (\text{total number of iterations}) \cdot \frac{1}{s} = t \cdot (\text{number of rigid-body transformations}), \quad (13)$$

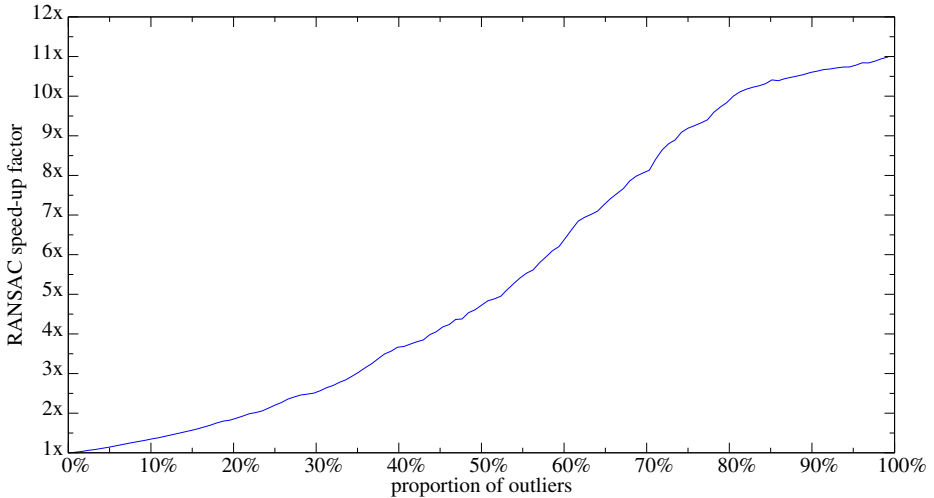


Fig. 6. RANSAC speed-up factor expected from the rigidity constraint

where t is the processing time needed for one iteration throughout the full homography evaluation process from four correspondences. This finally leads to:

$$s = \frac{(\text{total number of iterations})}{(\text{number of rigid-body transformations})}. \quad (14)$$

Based on the previous experimental results, this speed-up factor is represented in figure 6 with respect to the proportion of outliers. It shows that when the rigidity constraint is used, RANSAC is already twice as fast with only 22% of outliers. At 50% of outliers it is almost five times faster, and it still rises, becoming more than ten times faster with 80% of outliers. This shows that in any situation, the rigidity constraint not only prevents non rigid homographies, but also improves the overall performances of RANSAC.

5 Conclusion

In order to improve the results of the RANSAC algorithm in cases where the proportion of outliers is very large, an analysis of the invalid homographies obtained in such situations was performed in this paper. It led us to clarify the fact that a homography cannot be reduced to a rigid-body transformation and that the RANSAC algorithm is not always able to reject non rigid-body transformations. From these observations, a lightweight rigidity constraint has then been proposed, which makes it possible to prevent non-rigid-body transformations at a nearly negligible computing cost compared to the one needed for estimating a homography and checking all the putative correspondences for consistency with it. The evaluation tests have shown that a speed-up factor of more than ten can be expected in the presence of a large proportion of outliers. While the

discussion seems to be quite open concerning many evolutions of the original RANSAC algorithm [7], the impact of the presented rigidity constraint is objectively undeniable and profitable for any mapping algorithm and any RANSAC derivative.

Given that our first motivation was initially to prevent non rigid-body transformations when estimating homographies, the substantial speed-up factor achieved thanks to the proposed rigidity constraint is an unexpected, but very positive result.

References

1. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* 24, 381–395 (1981)
2. Faugeras, O.: *Three-Dimensional Computer Vision: a Geometric Viewpoint*. The MIT Press, Cambridge (1993)
3. Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*, 2nd edn. Cambridge University Press, Cambridge (2003)
4. Simler, C., Monnin, D., Georges, V., Cudel, C.: A robust technique to establish correspondences in case of important rotations around the optical axis. In: *Advanced Concepts for Intelligent Vision Systems, ACVIS 2004*, Brussels, Belgium (2004)
5. Simler, C., Monnin, D., Cudel, C., Georges, V.: Robust automatic image mosaic generation. In: *Proceedings of PSIP 2005, Fourth International Conference on Physics in Signal Image Processing*, Toulouse, France (2005)
6. Marquez-Neila, P., Miro, J.G., Buenaposada, J.M., Baumela, L.: Improving RANSAC for fast landmark recognition. In: *Computer Vision and Pattern Recognition Workshops*, pp. 1–8. IEEE Computer Society, Los Alamitos (2008)
7. Choi, S., Kim, T., Yu, W.: Performance evaluation of RANSAC family (2009)

Exploiting Neighbors for Faster Scanning Window Detection in Images

Pavel Zemčík, Michal Hradiš, and Adam Herout

Graph@FIT, Brno University of Technology, Bozotechnova 2, Brno, CZ
{zemcik,ihradis,herout}@fit.vutbr.cz

Abstract. Detection of objects through scanning windows is widely used and accepted method. The detectors traditionally do not make use of information that is shared between neighboring image positions although this fact means that the traditional solutions are not optimal. Addressing this, we propose an efficient and computationally inexpensive approach how to exploit the shared information and thus increase speed of detection. The main idea is to predict responses of the classifier in neighbor windows close to the ones already evaluated and skip such positions where the prediction is confident enough. In order to predict the responses, the proposed algorithm builds a new classifier which reuses the set of image features already exploited. The results show that the proposed approach can reduce scanning time up to four times with only minor increase of error rate. On the presented examples it is shown that, it is possible to reach less than one feature computed on average per single image position. The paper presents the algorithm itself and also results of experiments on several data sets with different types of image features.

1 Introduction

Scanning window technique is commonly used in object detection in images. In combination with highly selective and fast classifiers, it provides state-of-the-art success rates under real-time constraints for various classes of target objects [14,63]. Although, in reality, much information is shared between neighboring (overlapping) image positions, they are normally classified independently. Making use of this shared information has a potential to reduce amount of computations during scanning.

In this paper, we propose an effective and at the same time simple and computationally inexpensive method which uses the dependency between neighboring image position to suppress computing the original detection classifier at nearby locations. The proposed method learns a new classifiers which predict the responses of the original detection classifier at neighboring positions. When the prediction is confident enough, computing the original classifier is suppressed.

We propose to use WaldBoost algorithm [11] to learn the suppressing classifiers in such way that they reuse computations of the original detection classifier. These reused computations can be image features in case of Viola & Jones' [14]

like detectors or possibly also other temporal computation results. This reusing of computations is crucial and, in fact, is the only reason why faster detection can be achieved.

The task of learning the suppression classifiers is similar to emulating existing detectors by WaldBoost [12,13]. Formulating the neighborhood suppression task as detector emulation allows usage of unlabeled data for training and it does not require any modifications in learning of the detection classifier. Moreover, previously created detectors can be used without any modifications.

Although the classifiers proposed for scanning window detection vary highly, they also share many similarities which result from common requirements and similar properties of the target objects. The main requirements are high selectivity (low false alarm rate) and, in case of real-time processing, very low average classification time per position.

The classifiers generally rely on efficient image features to extract relevant information from the image. In literature, Haar-like features [14], Multi-block Local Binary Patterns [15], Local Rank Patterns [5], Histograms of Oriented Gradient (HOG) [3,4] and others have been shown to perform well in detection tasks.

Another common attribute of the detection classifiers is some form of focus-of-attention structure. The exact form of the attentional structure ranges from simple ad-hoc solutions [7] through more sophisticated [14,1] to theoretically sound approaches which minimize decision time for given target error rate on training data [11,2]. These attentional structures greatly reduce average classification time by rejecting most of the non-object positions early in the decision process. In attentional structures, the classifier is generally formed from several stages. After each of the stages a decision is made if it is already known with high enough confidence that the position does not contain the target object or further information has to be still extracted.

The previous approaches, which exploit the information shared by neighboring image positions in context of scanning window object detection, focus solely on sharing image features between classifiers computed at nearby locations. Schneiderman [10] advocates feature-centric computation of features as opposed to the commonly used window-centric evaluation. He proposes to compute simple discrete-valued features on a dense grid covering the whole image. These dense features are then used as input to efficiently implemented linear classifier. However, the feature-centric approach is suitable only early in the attentional classifiers. Schneiderman uses attentional cascade [14] where only the first stage is feature-centric and the rest is window-centric. The benefit of this approach vanishes when very fast classifiers are available (some detectors may need less than 2 features per position on average as shown in Section 3).

Dalal and Triggs [3] also use feature-centric computation of features. They use dense Histograms of Oriented Gradients image representation and a linear classifier trained by Support Vector Machine. This approach provides good detection rates for pedestrian detection; however, it is too computationally expensive to be used in real-time applications.

Except for the feature-centric evaluation, other ways to exploit the shared information are possible. Image features could be selected in such way that they are reused as much as possible when scanning the image. This approach, however, requires more complex learning methods. Alternatively, response of classifier at one position could be used as starting point (or as a feature) at neighboring location. Such access to previous results should provide good initial guess as the responses of classifiers at neighboring positions are highly correlated. However, this approach would also increase complexity of the learning system and would most likely require iterative retraining of the classifier which would significantly prolong the learning. On the the hand, the proposed approach of learning suppression classifiers can be used with existing detectors and the suppression classifiers are learned much faster than the original detector.

The suppression of some positions could be especially beneficial for some types of detectors and on certain computational platforms. If features that need normalization are used (e.g. Haar-like features and other linear features), suppressing some positions removes the need of possibly expensive computation of the local normalization coefficient. Also, on some platforms, the suppression could lead to faster execution as possibly deep computational pipeline does not have to be started for some positions.

The proposed neighborhood suppression method is presented in detail in Section 2 together with an algorithm able to learn the suppression classifiers. Results achieved by this approach are shown and discussed in Section 3. Finally, the paper is summarized and conclusions are drawn in Section 4.

2 Learning Neighborhood Suppression

As discussed before, we propose to learn classifiers suppressing evaluation of detection classifiers in the neighborhood of the currently examined image window. Such approach can improve detection speed only if the suppressing classifiers require very low overhead. This can be achieved by reusing computations already performed by the detection classifier itself. Most naturally, these reused computations can be responses of image features which are part of most real-time detectors [14,10,11,12,4,6,15,13]. In our work, the focus is only on these real-time detectors as they are the hardest to further speed up and speed of slower detectors can be improved by already known techniques [12,13].

The amount of information carried by the reused features, which is relevant to the decision task at neighboring location, will surely vary with different types of features and objects. It will also decrease with the distance of the two areas as the mutual overlap decreases.

In the further text, it is assumed that the detector for which the neighborhood suppressing classifier needs to be learned is a *soft cascade* [1]. This does not limit the proposed approach as extending it to detectors with different attentional structures is straightforward and trivial.

The soft cascade is a *sequential decision strategy* based on a *majority vote* of simple functions $h_t : \chi \rightarrow \mathbb{R}$ which are called *weak hypotheses* in the context of boosting methods [8]:

$$H_T(\mathbf{x}) = \sum_{t=1}^T (h_t(\mathbf{x})). \quad (1)$$

The weak hypotheses often internally operate with discrete values corresponding to partitions of the object space χ . Such weak hypotheses are called by Schapire and Singer [9] *space partitioning* weak hypotheses. Moreover, the weak hypotheses usually make their decision based only on a single image feature which is either discrete (e.g. LBP) or is quantized (e.g. Haar-like features and a threshold function). In the further text, such functions $f : \chi \rightarrow \mathbb{N}$ are referred to in general simply as *features* and the weak hypotheses are combinations of such features and a *look-up table functions* $l : \mathbb{N} \rightarrow \mathbb{R}$

$$h_t(\mathbf{x}) = l_t(f_t(\mathbf{x})). \quad (2)$$

In the further text, $c_t^{(j)}$ specifies the real value assigned by l_t to output j of f_t .

The decision strategy S of a soft cascade is a sequence of decision functions $S = S_1, S_2, \dots, S_T$, where $S_t : \mathbb{R} \rightarrow \{\#, -1\}$. The decision functions S_t are evaluated sequentially and the strategy is terminated with negative result when any of the decision functions outputs -1 . If none of the decision functions rejects the classified sample, the result of the strategy is positive.

Each of the decision functions S_t bases its decision on the tentative sum of the weak hypotheses H_t , $t < T$ which is compared to a threshold θ_t :

$$S_t(\mathbf{x}) = \begin{cases} \#, & \text{if } H_t(\mathbf{x}) > \theta_t \\ -1, & \text{if } H_t(\mathbf{x}) \leq \theta_t \end{cases}. \quad (3)$$

In this context, the task of learning a suppression classifier can be formalized as learning a new soft cascade with a decision strategy S' and hypotheses $h'_t = l'_t(f_t(\mathbf{x}))$, where the features f_t of the original classifier are reused and only the look-up table functions l'_t are learned.

2.1 Learning Suppression with WaldBoost

Soft cascades can be learned by several different algorithms [12]. We chose the *WaldBoost* algorithm [11,13] by Šochman and Matas which is relatively simple to implement, it guarantees that the created classifiers are optimal on the training data, and the produced classifiers are very fast in practice. The WaldBoost algorithm described in the following text is a slightly simplified version of the original algorithm. The presented version is specific for learning of soft cascades.

Given a weak learner algorithm, training data $\{(x_1, y_1) \dots, (x_m, y_m)\}$, $x \in \chi$, $y \in \{-1, +1\}$ and a target miss rate α , the WaldBoost algorithm solves a problem of finding such decision strategy that its miss rate α_S is lower than α and the average evaluation time $\bar{T}_S = E(\arg \min_i (S_i \neq \#))$ is minimal:

$$S^* = \arg \min_S \bar{T}_S, \text{ s.t. } \alpha_S < \alpha.$$

To create such optimal strategy, WaldBoost combines *AdaBoost* [9] and Wald's *sequential probability ratio test*. AdaBoost iteratively selects the most informative weak hypotheses h_t . The threshold θ_t is then selected in each iteration such that as many negative training samples are rejected as possible while asserting that the likelihood ratio estimated on training data

$$\hat{R}_t = \frac{p(H_t(\mathbf{x})|y = -1)}{p(H_t(\mathbf{x})|y = +1)} \quad (4)$$

satisfies $\hat{R}_t \geq \frac{1}{\alpha}$.

To learn the suppression classifiers we follow the classifier emulation approach from [13] which considers an existing detector a black box producing labels for new WaldBoost learning problem. However, when learning the suppression classifiers, the algorithm differs in three distinct aspects.

The first change is that when learning new weak hypothesis h'_t , only the look-up table function l'_t is learned, while the feature f_t is reused from the original detector. The selection of optimal weak hypothesis is generally the most time consuming step in WaldBoost and restricting the set of features thus makes learning the suppression classifier very fast.

The second difference is that the new data labels are obtained by evaluating the original detector on different image position than where the newly created classifier gets information from (the position containing the original features l_t). This corresponds to the fact that we want to predict response of the detector in neighborhood of the evaluated position.

The final difference is that the set of training samples is pruned twice in each iteration of the learning algorithm. As expected, samples rejected by the new suppression classifier must be removed from the training set. In addition, samples rejected by the original classifier must be removed as well. This corresponds to the behavior during scanning when only those features which are needed by the detector to make decision are computed. Consequently, the suppression classifiers can also use only these computed features to make their own decision. The whole algorithm for learning suppression classifier is summarized in Algorithm 1.

The neighborhood position is suppressed only when the suppression soft cascade ends with -1 decision. This way, the largest possible miss rate introduced by the suppression mechanism equals to α . The previous statement also holds when the detector is accompanied with multiple suppression classifiers which allows even higher sped-up still with controlled error.

Also, multiple neighboring position can be suppressed by a single classifier. Such behavior requires only slight change in Algorithm 1, where the training labels now become positive when the original detector gives positive result at any of the positions which should be suppressed.

2.2 Suppression in Real-Time Scanning Windows

The suppression with classifiers which reuse discrete-valued features is especially well suited for wide processor and memory architectures. On those architectures,

Algorithm 1. WaldBoost for learning suppression classifiers

Input: original soft cascade $H_T(x) = \sum_{t=1}^T h_t(x)$, its early termination thresholds $\theta^{(t)}$ and its features f_t ; desired miss rate α ; training set $\{(x_1, y_1) \dots, (x_m, y_m)\}$, $x \in \mathcal{X}$, $y \in \{-1, +1\}$, where the labels y_i are obtained by evaluating the original detector H_T at an image position with particular displacement with respect to the position of corresponding x_i

Output: look-up table functions l'_t and early termination thresholds $\theta'^{(t)}$ of the new suppression classifier

Initialize sample weight distribution $D_1(i) = \frac{1}{m}$

for $t = 1, \dots, T$

1. estimate new l'_t such that its

$$c_t^{(j)} = -\frac{1}{2} \ln \left(\frac{\Pr_{i \sim D}(f_t(x_i) = j | y_i = +1)}{\Pr_{i \sim D}(f_t(x_i) = j | y_i = -1)} \right)$$

2. add l'_t to the suppression classifier

$$H'_t(x) = \sum_{r=1}^t l'_r(f_r(x))$$

3. find optimal threshold $\theta'^{(t)}$
4. remove from the training set samples for which $H_t(x) \leq \theta^{(t)}$
5. remove from the training set samples for which $H'_t(x) \leq \theta'^{(t)}$
6. update the sample weight distribution

$$D_{t+1}(i) \propto \exp(-y_i H'_t(x_i))$$

multiple look-up tables l_t for a single feature f_t can be combined into single wide-word table such that single word contains $c_t^{(j)}$ values for all the classifiers. In such case, the required $c_t^{(j)}$ values can be loaded with single memory access, added to an accumulator register using single instruction and also efficiently compared with the rejection thresholds.

Obviously, such scheme is very well suitable for SIMD architectures, such as MMX/SSE instruction set extensions found in the contemporary PC processors. In such architectures, the wide registers can hold 4 32-bit numbers or 8 16-bit integer numbers. Consequently, the implementation of such scheme can be seen as nearly free of charge from the computational point of view.

The scheme is also applicable for programmable hardware or other hardware architectures. In such case, the scheme is beneficial in that addition of the extra prediction classifiers consumes only very little resources due to nearly unchanged structure and control subsystems.

On systems with wide enough data words but no SIMD support, the implementation can be similar to SIMD, except it must be assured that the multi-accumulator is not overflowed (as piecewise addition is not possible in this case). While this assumption seems to be severe and binding, the reality is such that

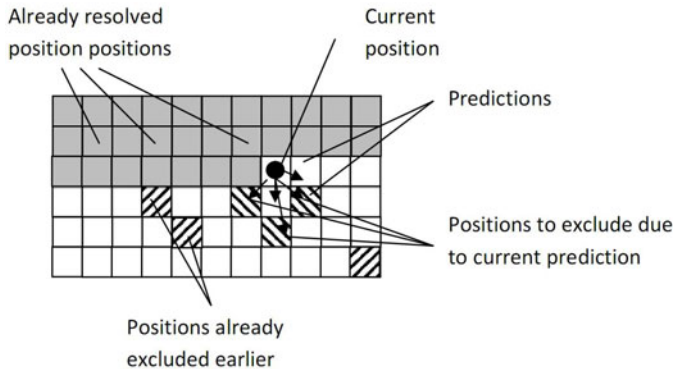


Fig. 1. Scanning an image in ordinary line-by-line fashion while using neighborhood suppression

Table 1. The benefit of neighborhood suppression for different features and datasets. ROCA is the percentage difference between area under ROC without and with area suppression. Time represents average number of features computed per position relative to the original detector without neighborhood suppression. "single" stands for suppressing single position. "12" stands for suppressing 12 positions with 12 suppression classifiers. Target error of the suppression classifiers was 5 %.

dataset	value	Haar		LBP		LRD		LRP	
		single	12	single	12	single	12	single	12
BioID	ROCA (%)	-0.02	0.07	-0.48	-3.44	-0.16	-1.08	-0.24	-2.04
	Time	0.96	0.68	0.78	0.33	0.92	0.54	0.82	0.37
PAL	ROCA (%)	-0.00	-0.39	-0.08	-0.21	-0.09	-0.85	-0.05	-0.44
	Time	0.96	0.71	0.77	0.31	0.91	0.51	0.82	0.36
CMU	ROCA (%)	-0.03	-0.36	-0.27	-1.92	-0.02	-0.49	-0.08	0.01
	Time	0.93	0.62	0.74	0.31	0.93	0.62	0.87	0.47
MS	ROCA (%)	-0.04	-0.54	-0.21	-1.02	-0.02	-0.27	-0.06	-0.65
	Time	0.93	0.60	0.73	0.29	0.93	0.60	0.87	0.45

it is easy to fulfill as the maximum possible value of each portion of the register can be calculated and predicted.

The suppression itself can be handled by binary mask covering positions to be scanned. The positions marked as suppressed are then excluded from further processing. The scanning order can remain the same as in ordinary scanning window approach, even though it restricts the positions which can be suppressed to those which are to the left and bottom of the currently classified position (see Figure 1). Possibly, more efficient scanning strategies can be developed, but such strategies are beyond the scope of this paper.

3 Experiments and Results

We tested the neighborhood suppression approach presented in the previous text on frontal face detection and eye detection. In both task, two separate test sets

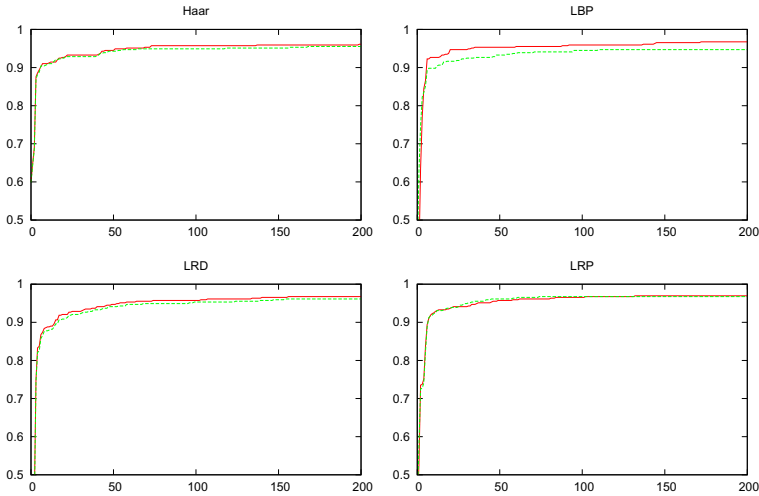


Fig. 2. The ROC curves on MIT+CMU dataset without suppression (full line) and with 12 suppression classifiers (dashed line). Target miss rate α of the suppression classifiers is 5 %.

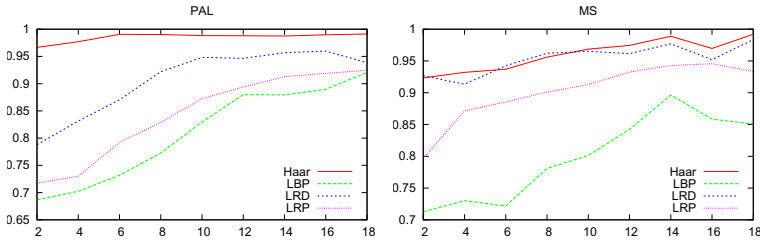


Fig. 3. Reduction of detection time (y-axis) when suppressing single positions in different horizontal distance from the classified position (x-axis). Target error of the suppression classifiers is 5 %.

were used - one with less constrained poses and lower quality images and one with easier poses and good quality images. For face detection, the harder dataset was standard MIT+CMU frontal face detection set (CMU) and the easier was a collection of 89 images of groups of people downloaded from the Internet. The easy set is denoted as MS and contains 1618 faces and 142M scanned positions. The eye detection classifiers were trained on XM2VTS¹ database and tested on BioID² database (104M positions, 3078 eyes) and on a easier dataset PAL³ (111M positions, 2130 eyes) which is similar to XM2VTS. When scanning, shift of the window was two pixels at the base detector resolution and scale factor was 1.2. The suppression classifiers were trained on a large set of unannotated images containing faces.

¹ <http://www.ee.surrey.ac.uk/CVSSP/xm2vtsdb/>

² <http://www.bioid.com/downloads/facedb/index.php>

³ <https://pal.utdallas.edu/facedb/>

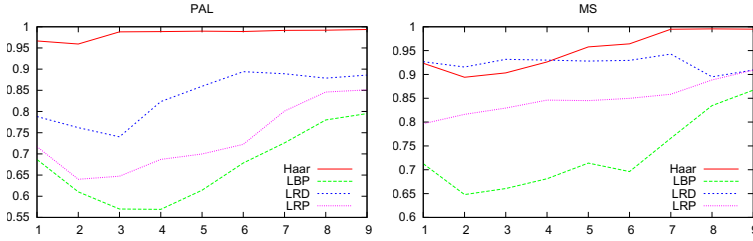


Fig. 4. Reduction of detection time (y-axis) when suppressing multiple positions on single image line by single classifier. x-axis is the number of suppressed positions. Target error of the suppression classifiers is 5 %.

The tests were performed with four types of image features which have been shown to perform well in real-time object detection. The features used were Haar-like features [14] (Haar), Multi-Block Local Binary Patterns [15] (LBP), Local Rank Differences [5] and Local Rank Patterns [5] (LRP). The real-valued responses of Haar-like features were normalized by standard deviation of local intensity and then quantized into 10 bins. The detection classifiers were learned by WaldBoost [11] algorithm and each contained 1000 weak hypotheses. The base resolution of the classifiers was 24 pixels wide.

In the first experiment, we focused on what is the the achievable speed-up using the neighborhood suppression of single and also twelve positions for moderately fast detection classifiers (4.5 - 6 features per position) and moderate target miss rate ($\alpha = 0.05$) and also on what is the influence of neighborhood suppression on precision of the detection. These results are shown in Table 1 and Figure 2. The results indicate large differences between individual image features. While the average number of weak hypotheses computed per position was reduced with twelve suppressed positions down to 30 % for LBP and 40 % for LRP, only 55 % was achieved for LRD and 65 % for Haar-like features. This can be explained by generally higher descriptive power of LBP and LRP. In general, the detection rate degraded only slightly with neighborhood suppression - by less than 1 % except for all twelve positions and LBP on datasets CMU and BioID and also LRP on BioID.

We have also evaluated the suppression ability with respect to distance form the classified position. Figure 3 shows that suppression ability decreases relatively slowly with distance and large neighborhood of radius at least 10 pixels can be used for the tested LBP and LRP classifiers.

As mentioned before, single suppression classifier can suppress larger area than just single position. Relation between speed-up and the size area of suppressed by a single classifier is shown in Figure 4. The results show that by suppressing larger area it is possible to reach higher speeds. However, the benefit is lower for frontal face detection and multiple suppression classifiers would always achieve higher speed-up.

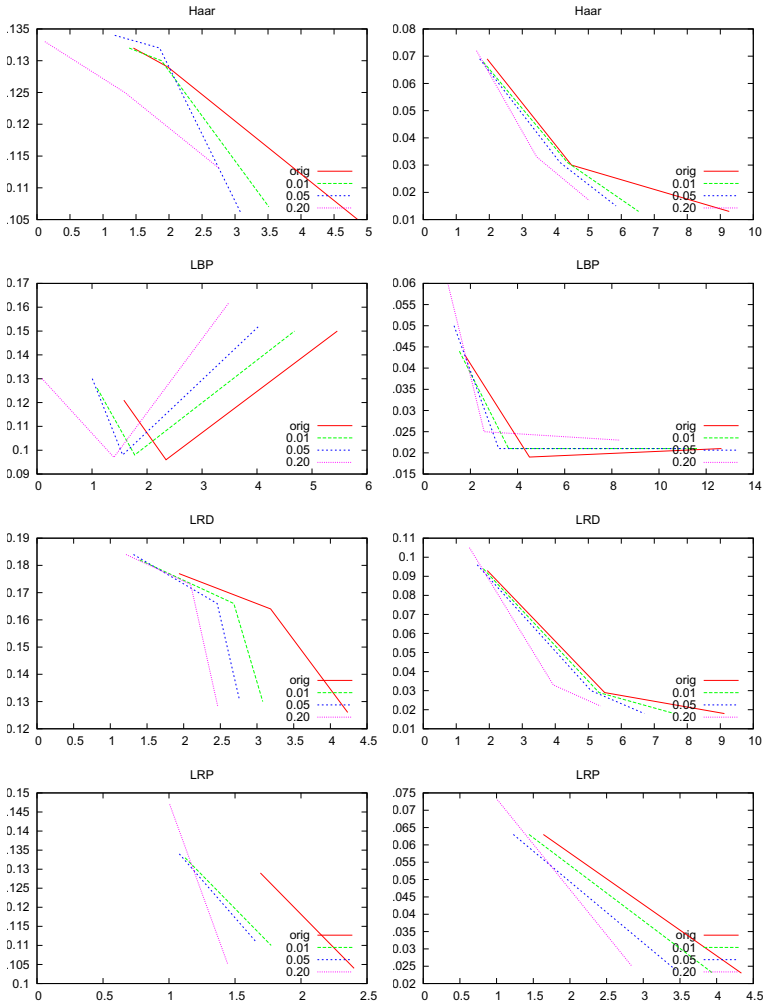


Fig. 5. Speed-up achieved by suppressing single position for different speeds of the original detector and different target miss rates α . Each line represents results for different α for three original detectors of different speed. X-axis is the speed of classifier in number of weak hypotheses evaluated on average per single scanned position (left is faster). Y-axis is area above ROC (lower is more accurate). On the left are results on eye detection PAL dataset and right are results on frontal face detection MS dataset.

For the neighborhood suppression to be useful, it must provide higher speed than simple detector for the same precision of detection. To validate this, we have trained number of detectors with different speeds (in terms of average number of features computed per position) for each feature type. Then, we learned three suppression classifiers with α set to 0.01, 0.05 and 0.2 for each of the detectors. The corresponding speeds and detection rates are shown in Figure 5. Even though, only a single suppression classifier is used in this case for each of the detectors, the results clearly show that by using neighborhood suppression, higher speed can be reached for the same detection rate.

4 Conclusions

This paper presents a novel approach to acceleration of object detection through scanning windows by prediction of the neighbor positions results using new classifiers that reuse the image features of the detector. The approach has been demonstrated on frontal face and eye detection using WaldBoost classifiers. The results clearly show that the proposed approach is feasible and that it can significantly speed up the detection process without loss of detection performance.

Further work includes evaluation of the approach on further data sets, other features, and possibly also different classification mechanisms, such as SVM. Further work will also focus on real-time implementation of the proposed method on CPU, GPU, and programmable hardware (FPGA). Also of interest will be possible improved image scanning patterns that can benefit even more from the neighborhood suppression.

Acknowledgements

This work has been supported by the Ministry of Education, Youth and Sports of the Czech Republic under the research program LC-06008 (Center for Computer Graphics) and by the research project "Security-Oriented Research in Informational Technology" CEZMSMT, MSM0021630528.

References

1. Bourdev, L., Brandt, J.: Robust object detection via soft cascade. In: CVPR (2005)
2. Cha, Z., Viola, P.: Multiple-instance pruning for learning efficient cascade detectors. In: NIPS (2007)
3. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), Washington, DC, USA, vol. 1, pp. 886–893. IEEE Computer Society Press, Los Alamitos (2005)
4. Hou, C., Ai, H.Z., Lao, S.H.: Multiview pedestrian detection based on vector boosting. In: Yagi, Y., Kang, S.B., Kweon, I.S., Zha, H. (eds.) ACCV 2007, Part I. LNCS, vol. 4843, pp. 210–219. Springer, Heidelberg (2007)

5. Hradiš, M., Herout, A., Zemk, P.: Local rank patterns - novel features for rapid object detection. In: Proceedings of International Conference on Computer Vision and Graphics 2008. LNCS, pp. 1–12 (2008)
6. Huang, C., Ai, H.Z., Li, Y., Lao, S.H.: High-performance rotation invariant multi-view face detection. *PAMI* 29(4), 671–686 (2007)
7. Rowley, H.A., Baluja, S., Kanade, T.: Neural network-based face detection. *IEEE Transactions On Pattern Analysis and Machine intelligence* 20, 23–38 (1998)
8. Schapire, R.E.: The boosting approach to machine learning: An overview. In: *MSRI Workshop on Nonlinear Estimation and Classification* (2002)
9. Robert, E.: Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions 37(3), 297–336 (1999)
10. Schneiderman, H.: Feature-centric evaluation for efficient cascaded object detection. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 29–36 (2004)
11. Sochman, J., Matas, J.: Waldboost - learning for time constrained sequential detection. In: *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, Washington, DC, USA, vol. 2, pp. 150–156. *IEEE Computer Society, Los Alamitos* (2005)
12. Sochman, J., Matas, J.: Learning a fast emulator of a binary decision process. In: Yagi, Y., Kang, S.B., Kweon, I.S., Zha, H. (eds.) *ACCV 2007, Part II. LNCS*, vol. 4844, pp. 236–245. *Springer, Heidelberg* (2007)
13. Sochman, J., Matas, J.: Learning fast emulators of binary decision processes. *International Journal of Computer Vision* 83(2), 149–163 (2009)
14. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, p. 511 (2001)
15. Zhang, L., Chu, R., Xiang, S., Liao, S., Li, S.Z.: Face detection based on multi-block lbp representation. In: *ICB*, pp. 11–18 (2007)

Optimisation-Based Image Grid Smoothing for SST Images

Guillaume Noel, Karim Djouani, and Yskandar Hamam

French South African Institute of Technology, Tshwane University of Technology,
Pretoria, South Africa

Abstract. The present paper focuses on smoothing techniques for Sea Surface Temperature (SST) satellite images. Due to the non-uniformity of the noise in the image as well as their relatively low spatial resolution, automatic analysis on SST images usually gives poor results. This paper presents a new framework to smooth and enhance the information contained in the images. The gray levels in the image are filtered using a mesh smoothing technique called SOWA while a new technique for resolution enhancement, named grid smoothing, is introduced and applied to the SST images. Both techniques (SOWA and grid smoothing) represent an image using an oriented graph. In this framework, a quadratic criterion is defined according to the gray levels (SOWA) and the spatial coordinates of each pixel (grid smoothing) and minimised using non-linear programming. The two-steps enhancement method is tested on real SST images originated from Meteosat first generation satellite.

Keywords: Grid smoothing, Graph-Based approach, Non-linear optimisation, SST, Remote sensing.

1 Introduction

The temperature of the ocean surface reflects important underlying oceanographic processes related to marine organisms and ecosystem dynamics. Areas of special interest due to their strong biological activity, thermal fronts are narrow regions of separation between two large areas of homogeneous temperature on the ocean surface. The water circulation associated with thermal fronts is responsible for the transportation system of the ocean. Oceanographers study these physical structures and create indices that interface the physical processes to the biological processes from which they can study the marine ecosystem and the marine fish population [1]. The behaviour of ocean mesoscale structures are usually modelled by a two layers ocean model, which ensures the continuity of the spatial derivatives of the temperature in at least one spatial direction [2]. The properties of the structures are not always depicted in the SST images due to the noise and the low spatial resolution of the image. Sea surface temperature (SST) images retrieved from satellites contain noise introduced by different atmospheric sources that complicates automatic detection. Clouds absorb infrared emission and limit the information that is available on each SST image [3]. The

strength of the wind is also affecting the measurement of the SST by the satellite sensor. These reasons lead to a non-uniform repartition of the noise properties in the image. The spatial resolution of the SST images are typically 4km by 4km. Knowing that the difference in temperature may peak up in certain region of the world to 2 degrees per km, the shape of the structures are not conserved in the image. Various methods have been used to filter the SST images including low-pass filter [4], contextual filter [1], adaptive filtering [5] and others [6]. Some of the methods try to address the non-uniformity of the noise while the other are focusing on the shape of the structures. This paper presents a common framework to tackle both issues at the same time using a common formulation of the problem and mathematical tools to solve it. Previous work on the grid smoothing or interpolation can be found in [7] where the image is modelled as a non-resistive or resistive power grid, in [8] where strong constraints on the shape of the object are assumed, and in [9], where hierarchical grid construction is introduced. Previous interesting work on interpolation of large dataset using optimisation techniques may be found in [10] and [11] where a weighting factor between the model and the data terms is introduced. The framework presented in the present paper is twofold. In the first step, the SOWA algorithm introduced in [12] is applied to the image to remove the noise. The SOWA algorithm uses the mesh representation of an image and a quadratic cost function is defined with the gray levels present in the image. The minimisation of the cost function leads to a new set of gray levels preserving the shape of the objects in the image while reducing the level of noise. The second step, called grid smoothing, tackles the issue of the low resolution of the SST images. Using the mesh representation of the image and non-linear programming, the initial uniform grid on which the image is sampled is modified to fit the content. The result of the grid smoothing is a non-uniform grid exposing more points in the region of large variance. Section 2 introduces the mesh representation of the image used in the present paper. Section 3 reviews the optimisation-based approach to mesh smoothing (SOWA) while Section 4 introduces the grid smoothing framework. The results are presented and discussed in Section 5. Section 6 summarizes the contribution of the present paper and discusses recommendations and the future works.

2 Graph-Based Image Representation

2.1 First Order Node-Edge Matrix

Our input data is a graph $G = (V, E)$, embedded in the 3D Euclidian space. Each edge e in E is an ordered pair (s, r) of vertices, where s (resp. r) is the sending (resp. receiving) end vertex of e . To each vertex v is associated a triplet of real coordinates x_v, y_v, z_v [12]. Let C be the node-edge incidence matrix of the graph G , defined as:

$$C = \begin{cases} 1 & \text{if } v \text{ is the sending end of edge } e \\ -1 & \text{if } v \text{ is the receiving end of edge } e \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

If $A = C^t C$, as C^t is not full ranked (sum of rows is equal to zero), the determinant of A is zero. Furthermore, let $z = C y$, then have $y^t C^t C y = z^t z \geq 0$ and hence A is positive semi-definite. The matrix A matrix is usually sparse for large problems, with the diagonal elements $a_{ij} =$ number of edges incidents to vertex i ; and the off-diagonal elements:

$$a_{ij} = \begin{cases} -1 & \text{if an edge exists between vertices } i \text{ and } j \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

In the literature, this matrix is referred to as the Laplacian matrix (also called topological or graph Laplacian), it plays a central role in various applications.

If $\tilde{A} = |A \cdot \Psi|$, where \cdot is the elementwise matrix multiplication operator (or Hadamard-Schur product) and

$$\Psi = \begin{pmatrix} 0 & 1 & & & 1 \\ 1 & 0 & 1 & & \\ & 1 & 0 & 1 & \\ 1 & & \ddots & \ddots & \ddots \end{pmatrix} \tag{3}$$

\tilde{A} is commonly known as the adjacency matrix of the graph.

2.2 N^{th} Order Node-Edge Matrix

C as defined below represents the first order connectivity of the image. In many cases, it might also be useful to define an extended node-edge matrix taking into account "weak" connection between nodes. For example, we might be interested to define a "weak" connection between second order neighbouring nodes. C can be then defined by:

$$C_{1..N} = \begin{bmatrix} C_1 \\ \vdots \\ C_N \end{bmatrix} \tag{4}$$

where C_1 represents the connection (between first order neighbours) and C_N represents the connection (between N^{th} order neighbours).

$$A = C_{1..N}^t C_{1..N} = \begin{bmatrix} C_1 \\ \vdots \\ C_N \end{bmatrix}^t \cdot \begin{bmatrix} C_1 \\ \vdots \\ C_N \end{bmatrix} = [C_1^t C_1 + \dots + C_N^t C_N] \tag{5}$$

Like in the definition of the node-edge matrix, the coefficients of C_k are either equals to 0, c_k or $-c_k$. The c_k coefficients are strictly positive and have to verify the following properties:

$$\begin{cases} 1 \geq c_k \\ c_k \geq c_{k+1} \end{cases} \tag{6}$$

For example, the c_k coefficients can be defined by the following recurrence relation:

$$\begin{cases} c_1 = 1 \\ c_{k+1} = \frac{1}{2}c_k \end{cases} \quad (7)$$

If we consider an image which size is $N \times N$ pixels, the number of non-null elements in C_1 is equal to $2N(N - 1)$ if we assume a four pixels connectivity. The number of non-null elements in C_2 is proportional to N^2 . It is obvious that for computational reason, we will not consider connection of higher degree than two.

2.3 Notation

Considering an image with M pixels, in the following section, X , Y and Z will respectively represent $[x_1, \dots, x_M]^t$, $[y_1, \dots, y_M]^t$ and $[z_1, \dots, z_M]^t$. X and Y are uniformly distributed (coordinates of the pixels in the plane), while Z represents the value of the pixels. Each pixel in the image is numbered according to its column and then its rows. For a square image, $M = N^2$, N being the number of pixel in a row (or column). When C is used, it implicitly represents the first order node-edge matrix. The notation $C_{1..N}$ will be used for higher order node edge matrix.

3 Optimisation-Based Approach to Mesh Smoothing

The present section presents an overview of the method. The idea is to generalize and reformulate Laplacian smoothing. A detailed approach can be found in [12].

3.1 General Framework

Hamam and Couprie showed in [12] that mesh smoothing may be reformulated as a minimisation of the cost function J as defined below:

$$J = \frac{1}{2} \left[\left(Z - \hat{Z} \right)^t Q \left(Z - \hat{Z} \right) + \theta_0 Z^t Z + \theta_1 Z^t \bar{A} Z + \theta_2 Z^t \bar{A}^2 Z \right] \quad (8)$$

where

- Q is a symmetric positive definite weighing matrix,
- θ_0, θ_1 and θ_2 are weighing scalars,
- $\bar{A} = C^t \Omega C$, and Ω is a diagonal matrix of weight associated to each edge,
- C is the node-edge matrix of the image,
- Z and \hat{Z} are respectively the value of the pixels and their initial value.

The inclusion of initial values in the cost function prevents the smoothing from shrinking the object. For large size problem, a gradient descent algorithm may be used to minimise J and the convergence is guaranteed.

3.2 Second Order Algorithm with Attach (SOWA)

In the SOWA algorithm, the cost function J can be expressed as follows:

$$J = \frac{1}{2} \left[(Z - \hat{Z})^t Q (Z - \hat{Z}) + \theta Z^t \bar{A}^2 Z \right] \quad (9)$$

For small size problem, the solution can be found by matrix inversion,

$$Z_{opt} = (I + \theta \bar{A}^2)^{-1} \hat{Z} \quad (10)$$

and the solution is unique. For large size problems, the gradient descent method may be applied. One iteration of the gradient descent method is as follows:

$$Z^{n+1} = Z^n - \alpha^n \nabla_x J = Z^n - \alpha^n ((Z^n - Z) + \theta \bar{A}^2 Z^n) \quad (11)$$

where n is the iteration number and α^n is a positive scalar corresponding to the step in the opposite direction of the gradient. The SOWA method is chosen when the smoothing needs to conserve the curves of the image. The SOWA algorithm is applied to Meteosat first generation satellite and the results are depicted in the results section of the paper. It may be noted that the SOWA algorithm is only used to filter the gray levels of the image to reduce the level of noise in the images. It may be shown that this method is more efficient than most of usual adaptive filtering techniques on our dataset.

4 Optimisation-Based Approach to Grid Smoothing

The goal of the grid smoothing applied to large scale SST images is to enhance the resolution. The initial uniform grid on which the image is sampled is modified to fit the content of the image. After modification of the grid using the grid smoothing approach, the regions with large variance values expose a greater number of points of the grid while the opposite phenomenon may be seen in the region with small variance. It may be noted that the total number of points remains unchanged.

4.1 General Framework

A cost function is introduced to adapt the grid to the information contained in the image. A cost function J is defined as follows:

$$J = J_X + J_Y \quad (12)$$

where

$$J_X = \frac{1}{2} \left[(\bar{Z} \cdot \bar{Z})^t (\bar{X} \cdot \bar{X}) + \theta (X - \hat{X})^t Q (X - \hat{X}) \right] \quad (13)$$

and

$$J_Y = \frac{1}{2} \left[(\bar{Z} \cdot \bar{Z})^t (\bar{Y} \cdot \bar{Y}) + \theta (Y - \hat{Y})^t Q (Y - \hat{Y}) \right] \quad (14)$$

where \cdot represents the element-wise matrix multiplication and $\bar{X} = CX, \bar{Y} = CY$ and $\bar{Z} = CZ$.

θ is a real number and can be seen as a normalisation factor. A convenient choice for θ can be:

$$\theta = \| CZ \|_2 \tag{15}$$

where $\| \cdot \|_2$ represents the L_2 norm of the vector CZ .

J_X and J_Y are the sum of two terms. The first one will minimise the sum of the surfaces of the triangles formed by two connected points and the projection of one of the point on the Z-axis. As a result, the density of the points in the area where the variations in Z are large will increase. The second terms $(\theta (X - \hat{X})^t Q (X - \hat{X}))$ or $\theta (Y - \hat{Y})^t Q (Y - \hat{Y}))$ is a weighting in respect of the initial coordinates of the points to avoid large movement in the grid.

4.2 Convergence

Consider the optimisation problem of the function J_X (the convergence of J_Y can be proven in a similar fashion). The element-wise product of \bar{X} by \bar{X} can be written as follows:

$$\bar{X} \cdot \bar{X} = \left(\sum G^i \bar{X} (g^i)^t \right) \bar{X} \tag{16}$$

where G^i is a square matrix which elements are null except the i^{th} element of the diagonal which is equal to 1 and g^i is a vector which elements are null except the i^{th} which is equal to 1.

If $(u^i)^t = (\bar{Z} \cdot \bar{Z})^t G_i$, J_X can be expressed as

$$J_X = \frac{1}{2} \left(\sum (u^i)^t \bar{X} (g^i)^t \bar{X} + \theta (X - \hat{X})^t (X - \hat{X}) \right) \tag{17}$$

The gradient is then equal to

$$\nabla_x J_X = \sum u^i (g^i)^t CX + X - \hat{X} \tag{18}$$

At the optimum, $\nabla_x J_X = 0$

$$\sum u^i (g^i)^t CX_{opt} + X_{opt} - \hat{X} = 0 \tag{19}$$

$$\left(\sum u^i (g^i)^t C + I \right) X_{opt} = \hat{X} \tag{20}$$

The inverse of $\left(\sum u^i (g^i)^t C + I \right)$ exists and for small size problems the above equation may be solved to give

$$X_{opt} = \left(\sum u^i (g^i)^t C + I \right)^{-1} \hat{X} \tag{21}$$

For large size problem, the inversion of the matrix is computationally too expensive. The gradient descent method can then be applied. The gradient descent algorithm can be written as

$$X^{n+1} = X^n - \alpha^n \nabla_x J_X = X^n - \alpha^n \left(\sum u^i (g^i)^t CX + X - \hat{X} \right) \quad (22)$$

where n is the iteration number and α^n is a positive scalar corresponding to the step in the opposite direction of the gradient.

4.3 Grid Smoothing with Linear Equality Constraints

The adaptation of the grid presented in the previous section does not ensure that the initial size of the image remains the same. In broader terms it can be useful in various cases to fix the coordinates of some points of the image.

The linear constraints can be expressed as

$$(X - \hat{X})^t B = 0 \quad (23)$$

where B is a vector whose size is the total number of points in the image and the values of B verify the following properties

$$B = \begin{cases} 1 & \text{if the point belongs to } \Phi \\ 0 & \text{otherwise} \end{cases} \quad (24)$$

where, Φ is the set of points whose coordinates will remain unchanged.

Using the properties of the vector B , the grid smoothing with fixed points can be formalized as a nonlinear optimization problem with linear constraints as follows:

$$\begin{cases} \text{minimise } J(X, Y) \\ (X - \hat{X})^t B = 0 \end{cases} \quad (25)$$

The optimization problem can be solved using the Lagrangian parameters. Fixing certain points in the image can be convenient if, for example, the size of the image needs to be unchanged. In this case, B would be equal to the concatenated rows of the following matrix Φ :

$$\Phi = \begin{pmatrix} 1 & 1 & \cdots & 1 & 1 \\ 1 & 0 & \cdots & 0 & 1 \\ \vdots & \vdots & & \vdots & \vdots \\ 1 & 0 & \cdots & 0 & 1 \\ 1 & 1 & \cdots & 1 & 1 \end{pmatrix} \quad (26)$$

4.4 Grid Smoothing with Inequality Constraints

As described in the sections above, the points of the grid can move according to the change in temperature. However, the connection between the points remains the same. A constraint about the region of the plane where the points can move has to be introduced to make sure that the optimization will not end up with a graph containing intersecting connections. The constraint can be expressed as

$$\begin{cases} CX \leq 0 \\ CY \leq 0 \end{cases} \quad (27)$$

Another type of constraint could be to limit the movement of the points. In this case, the constraint could be express as

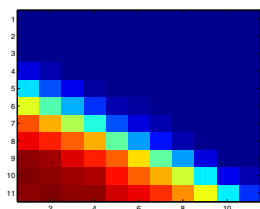
$$\begin{cases} X - \hat{X} \leq \epsilon \\ Y - \hat{Y} \leq \epsilon \\ 0 \leq \epsilon \leq 0.5 * step \end{cases} \quad (28)$$

with *step* being the step of the initial grid.

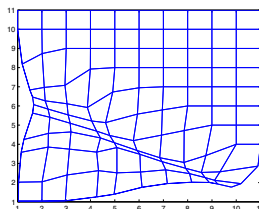
5 Simulation Results

The image dataset used in the experiments is originated from by Meteosat satellite. The SST images result from an average of the sea surface temperature collected over a month. The first set of experiments aims at investigating the effect of the neighbouring order and the constraints in the grid smoothing while the second set exposes the results of the complete image processing chain (SOWA followed by grid smoothing). The non-linear optimization method used in both SOWA and grid smoothing is the conjugate gradient. The typical convergence time is around 1 second for a $100pixels \times 100pixels$ image.

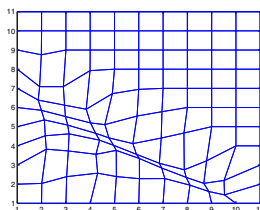
Figure [1](#) focuses on a detail of an SST image ($50km \times 50km$) including a thermal front. As explained previously, a front characterizes the transition between two regions of homogeneous temperature and may be interpreted as an edge. The two homogeneous regions depict small variance, while the front itself is characterized by a large variance. It may be observed in the results that the grid smoothing has no or little effect on the homogeneous region (the grid stays quasi-uniform) while the front itself depicts a larger number of points. In the grid smoothing process, the edges act like attractors for the points in the grid. When the optimisation process is unconstrained, the dimensions of the new grid do not match the initial boundaries of the image, which may lead to geometrical issues while reconstructing the image. However, it may be observed that constraining the grid is leading to a loss of accuracy in the boundary regions. Comparing the two orders of grid smoothing, it may be observed that a greater shrinkage of the image is seen in the second order compared to the first order smoothing. The grid is also denser in the second order in the regions where the temperature is changing. As the second order smoothing uses a second order neighbourhood between the points, the presence of an edge not only attracts its direct neighbours but also further points in the grid. The computational cost of the second order compared to the first order is, however, a major drawback. A trade-off between the accuracy and the computing time is to be found according to the application.



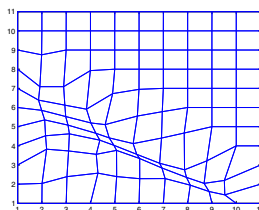
(a) Detail of SST image (Me-teosat)



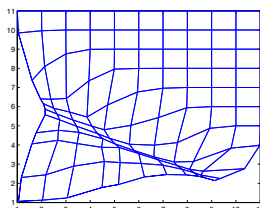
(b) Smoothed grid without constraints (first order)



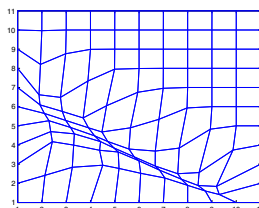
(c) Smoothed grid with equality constraints (first order)



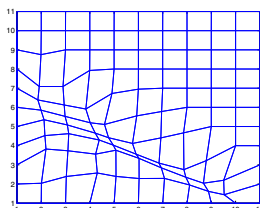
(d) Smoothed grid with equality and non-equality constraints (first order)



(e) Smoothed grid without constraints (second order)

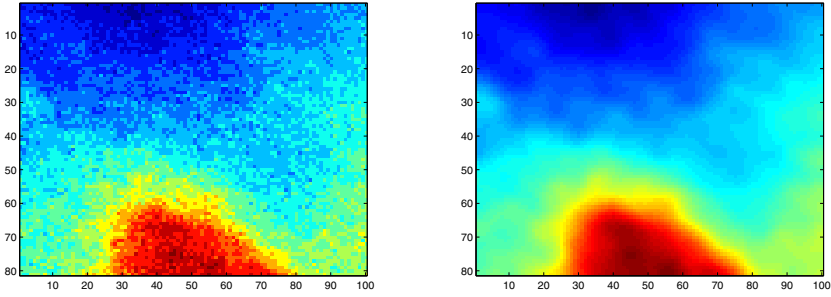


(f) Smoothed grid with equality constraints (second order)



(g) Smoothed grid with equality and non-equality constraints (second order)

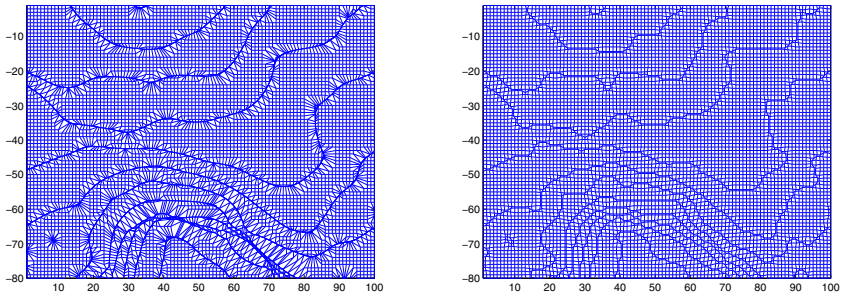
Fig. 1. First and second order grid smoothing



(a) SST Image (Meteosat, monthly average, July 2003)

(b) Filtered image using SOWA

Fig. 2. Result of the mesh smoothing



(a) Grid smoothing with $\theta = 30$

(b) Grid smoothing with $\theta = 0.3$

Fig. 3. Result of the grid smoothing

Figure 3 and 2 displays respectively the initial SST image, the smoothed image using the SOWA algorithm and the results of the grid smoothing for two values of θ . It is obvious that the level of noise in the initial image is relatively high. The sources of the noise are the ones mentioned before (wind at the surface of the sea and cloud coverage) plus in our database the effect of the averaging process. The evolution of the phenomena observed at the surface of the ocean (fronts, eddies...) is relatively fast. Associated with the cloud coverage, the number of available observations of each pixel in the image is not the same and may be small. On the other hand, as the meso-scale structures are moving, an average on a month only allows the researcher to observe the large scale slow moving evolution of the structures. A spatial smoothing is also performed on the shape of the sea structures. The present SST image represents a region in the southern Indian Ocean, approximately 500km south of the city of Durban, South Africa. The dimensions of the region depicted is about 300km * 400km.

The SOWA algorithm is applied on the SST image and it may be seen that the level of noise in the image decreases. However, from a qualitative point of

view, the content in the image looks preserved. The efficiency of this type of mesh smoothing has been proven on our dataset and its performance may be compared favourably to the other filtering techniques for satellite images.

The grid smoothing process is applied on the smoothed image and the results are depicted in Figure 2a and Figure 2b. It may be seen that the concentration of points in the region presenting an edge (thermal front) is greater than in the other regions. As θ increases, the deformation of the grid increases, the weight of the initial coordinates being decreased in the cost function. With a large value of θ , the repartition of the points is smoother along the edges while the details in the shape are lost. On the other hand, a small value of θ leads to greater details in the shape recovered, at the expense of a sparser repartition of the points. In any case, continuous region with a large number of points may be observed and may be interpreted as the boundaries of meso-scale sea structures.

6 Conclusion

A common framework for data smoothing and grid smoothing was presented in the paper. A cost function was introduced for each case and that the solution of the minimisation is unique. Using the conjugate gradient method, the computing time is reasonable and large image may be processed. An extensive study of the convergence and computing time of the method may be found in [13]. Multiple applications of the framework are possible and will be investigated in future research. Improved edge detection, image enhancement and compression are among them. The reconstruction of the image will also be investigated and compared to other interpolation schemes like in [10] and [11]. Finally, the enhanced images will be fed into a variational data assimilation scheme (4D-Var for example) to test their ability to forecast the evolution of the sea surface temperature.

References

1. Belkin, I.M., O'reilly, J.E.: An algorithm for oceanic front detection in chlorophyll and SST satellite imagery. *Journal of Marine Systems* 78(3), 319–326 (2009)
2. Huot, E., Herlin, I., Korotaev, G.: Assimilation of SST satellite images for estimation of ocean circulation velocity. In: *Geoscience and Remote Sensing Symposium*, pp. II847–II850 (2008)
3. Cayula, J.-F., Cornillon, P.: Cloud detection from a sequence of SST images, *Remote Sens. Environ.* 55, 80–88 (1996)
4. Hai, J., Xiaomei, Y., Jianming, G., Zhenyu, G.: Automatic eddy extraction from SST imagery using artificial neural network. In: *Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science*, Beijing (2008)
5. Lim, Jae, S.: *Two-Dimensional Signal and Image Processing*, p. 548. Prentice Hall, Englewood Cliffs (1990) equations 9.44 – 9.46

6. Guindos-Rojas, F., Canton-Garbin, M., Torres-Arriaza, J.A., Peralta-Lopez, M., Piedra Fernandez, J.A., Molina-Martinez, A.: Automatic Recognition of Ocean Structures from Satellite Images by Means of Neural Nets and Expert Systems. In: Proceedings of ESA-EUSC 2004 - Theory and Applications of Knowledge-Driven Image Information Mining with Focus on Earth Observation (ESA SP-553), Madrid, Spain, March 17-18 (2004)
7. Jiang, F., Shi, B.E.: The memristive grid outperforms the resistive grid for edge preserving smoothing, *Circuit Theory and Design*. In: ECCTD 2009, pp. 181–184 (2009)
8. Bu, S., Shiina, T., Yamakawa, M., Takizawa, H.: Adaptive dynamic grid interpolation: A robust, high-performance displacement smoothing filter for myocardial strain imaging. In: Ultrasonics Symposium, IUS 2008, November 2-5, pp. 753–756. IEEE, Los Alamitos (2008)
9. Huang, C.-L., Hsu, C.-Y.: A new motion compensation method for image sequence coding using hierarchical grid interpolation. *IEEE Transactions on Circuits and Systems for Video Technology* 4(1), 42–52 (1994)
10. Stals, L., Roberts, S.: Smoothing large data sets using discrete thin plate splines. *Computing and Visualization in Science* 9, 185–195 (2006)
11. Roberts, S., Stals, L.: Discrete thin plate spline smoothing in 3D. *ANZIAM Journal* 45 (2003)
12. Hamam, Y., Couprie, M.: An Optimisation-Based Approach to Mesh Smoothing: Reformulation and Extension. In: Torsello, A., Escolano, F., Brun, L. (eds.) GbRPR 2009. LNCS, vol. 5534, pp. 31–41. Springer, Heidelberg (2009)
13. Noel, G., Djouani, K., Hamam, Y.: Grid smoothing: A graph-based approach. In: Cesar Jr., R.M. (ed.) CIARP 2010. LNCS, vol. 6419, pp. 183–190. Springer, Heidelberg (2010)

Estimating 3D Polyhedral Building Models by Registering Aerial Images

Fadi Dornaika^{1,2} and Karim Hammoudi³

¹ University of the Basque Country, San Sebastian, Spain

² IKERBASQUE, Basque Foundation for Science, Bilbao, Spain

³ Université Paris-Est, Institut Géographique National, Paris, France

Abstract. We describe a model driven approach for extracting simple 3D polyhedral building models from aerial images. The novelty of the approach lies in the use of featureless and direct optimization based on image rawbrightness. The 3D polyhedral model is estimated by optimizing a criterion that combines a global dissimilarity measure and a gradient score over several aerial images. The proposed approach gives more accurate 3D reconstruction than feature-based approaches since it does not involve intermediate noisy data (e.g., the 3D points of a noisy Digital Elevation Model). We provide experiments and evaluations of performance. Experimental results show the feasibility and robustness of the proposed approach.

1 Introduction and Motivation

The extraction of 3D models of buildings from aerial images is currently a very active research area since it is a key issue in urban planning, virtual reality, and updating databases for geo-information systems, to name a few [1]. The proposed methods for building reconstruction differ by the assumption made as well as by the type of input data. However, one can easily classify these approaches into two main categories: bottom-up and top-down approaches. In theory, bottom-up approaches can handle the case where there is no prior knowledge about the sought building model. However, in the presence of noisy or low resolution data there is no guarantee that the estimated models will be correct. On the other hand, top-down approaches rely on some prior knowledge (e.g., using parametric models). The top-down approaches use the principle of hypothesis-verification in order to find the best model fit. Both categories have been used with features that are extracted and matched in at least two images. For roofs, the most used image features are 2D segments and junctions lines that are converted into 3D features. The final polyhedral model is then estimated from these 3D features. Model-based reconstruction techniques were first applied in digital photogrammetry for the (semi-)automatic reconstruction of buildings in aerial images with the help of generic building models [2-4]. In this paper, we propose a featureless approach that extracts simple polyhedral building models from the rawbrightness of calibrated aerial images where the footprint of the building in one image

is obtained either manually or automatically [5]. We were inspired by the featureless image registration techniques where the goal is to compute the global motion of the brightness pattern between them (e.g., affine or homography transforms) without using matched features [6].

Unlike existing approaches for building reconstruction, our approach derives the polyhedral building model by minimizing a global dissimilarity measure based on the image rawbrightness. It is carried out using a genetic algorithm. To the best of our knowledge the use of featureless and direct approaches has not been used for extracting polyhedral models of buildings. In any feature-based approach, the inaccuracies associated with the extracted features, in either 2D or 3D, will inevitably affect the accuracy of the final 3D model. Most of the feature-based approaches use sparse extracted features such as interest points and line segments. Thus, the sparseness of data coupled with noise will definitely affect the accuracy of the final building reconstruction.

Recently, many researchers proposed methods for extracting polyhedral models from Digital Elevation Models (DEMs) (e.g., [3, 7, 8]). Compared to these approaches, our method has the obvious advantage that the coplanarity constraints are implicitly enforced in the model parametrization. On the other hand, the approaches based on DEMs impose the coplanarity constraint on the 3D points of the obtained surface in the process of plane fitting. DEMs are usually computed using local correlation scores together with a smoothing term that penalizes large local height variation. Thus, correlation-based DEMs can be noisy. Moreover, height discontinuities may not be located accurately. In brief, our proposed approach can give more accurate 3D reconstruction than feature-based approaches since the process is more direct and does not involve intermediate noisy data (e.g., the 3D points of a noisy DEM).

Although the proposed method can be used without any DEM it can be useful for rectifying the polyhedral models that are inferred from DEMs. In this case, our proposed method can be useful in at least two cases. The first case is when the provided model is erroneous, e.g., a facet is not modelled. Figure 1 illustrates two corresponding examples of erroneous estimated polyhedral models. The second case is when the estimated shape is correct but its geometric parameters are not accurate enough, e.g., the coordinates of some vertices are not very precise. In the latter case our proposed approach can be used for improving the accuracy of the model parameters. The remainder of the paper is organized as follows. Section 2 states the problem we are focusing on and describes the parametrization of the adopted polyhedral model. Section 3 presents the proposed approach. Section 4 gives some experimental results.

2 Problem Statement

Since aerial images are used only roof models can be estimated. In this work, we restrict our study to simple polyhedral models that are illustrated in Figure 2. The model illustrated in Figure 2(a) can describe a building roof having two, three, or four facets. This is made possible since the 3D coordinates of the

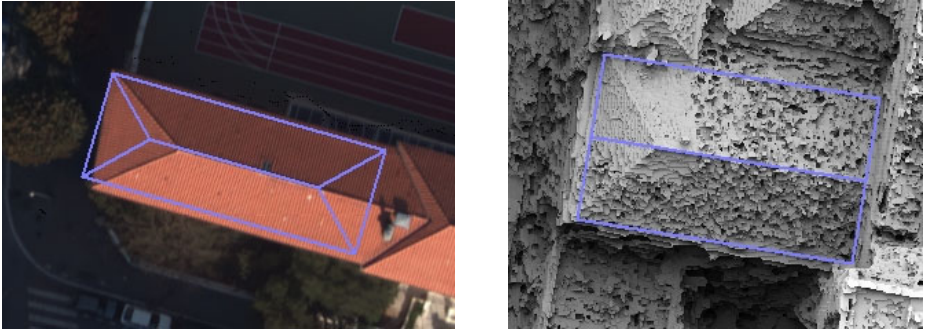


Fig. 1. Two examples of erroneous estimated 3D polyhedral models that were inferred from Digital Elevation Models (DEMs). Left: the estimated model has four facets while the real roof is composed of three facets. Right: the estimated model has two facets while the real roof is composed of three facets.

inner vertices are considered as unknown. These models can describe all typical situations: non symmetric shape, sloping ground or roofs (i.e., every vertex can have a different height). Because a complex building can be described as an aggregation of simple building models, our approach can also deal with complex buildings once a partitioning of the building into simple building-parts is done. However, dealing with complex buildings is beyond the scope of the paper.

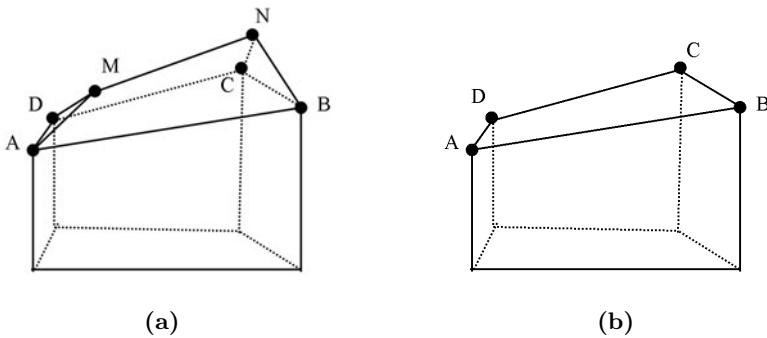


Fig. 2. The adopted simple polyhedral models. (a) The multi-facet model. (b) The one facet model.

The problem we are focusing on can be stated as follows. Given the footprint of a building in one aerial image we try to find the polyhedral model (an instance of the models depicted in Figure 2) using the rawbrightness of the aerial images that views this building. The flowchart of the proposed approach is depicted in Figure 3. Since the images are calibrated (the camera intrinsic parameters are known) and since the 2D locations of the outer vertices are known in one image, our simple polyhedral (Figure 2.a) model can be fully parameterized by

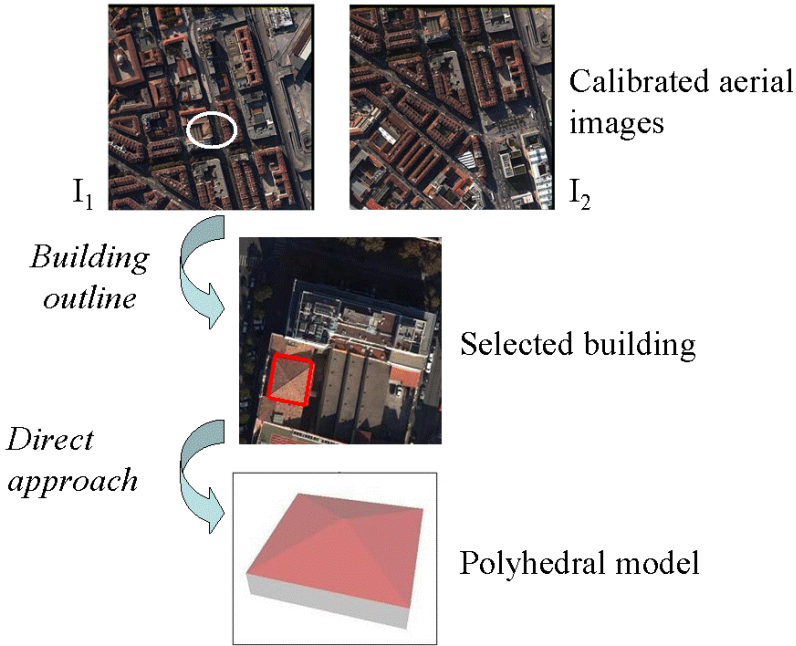


Fig. 3. Extracting 3D polyhedral models from image rawbrightness

eight parameters: four parameters for the 2D location of the inner vertices M and N and four parameters for the height of the vertices A , M , N , and C . The remaining vertices are determined by intersecting the corresponding line of sight with the estimated support planes. The eight parameters are encapsulated into one single vector \mathbf{w} :

$$\mathbf{w} = (U_M, V_M, U_N, V_N, Z_A, Z_M, Z_N, Z_C)^T \tag{1}$$

where (U_M, V_M) and (U_N, V_N) are the image coordinates of the vertices M and N , respectively. Recall that the 3D coordinates are expressed in a local coordinate system whose Z axis coincides with the ground normal since the aerial images are geo-referenced. In practice, although the 2D location of the inner vertices is not known, the 2D line (the projection of a ridge segment) going through them can be easily extracted from the master image by using a simple edge detector followed by a Hough transform. Once the equation of this 2D line is known, the parametrization of the building model can be simplified to:

$$\mathbf{w} = (\lambda_M, \lambda_N, Z_A, Z_M, Z_N, Z_C)^T \tag{2}$$

where λ_M and λ_N parameterize the 2D location of the inner vertices along the 2D segment obtained by intersecting the 2D line with the building footprint. Thus, finding the model boils down to finding the vector \mathbf{w} .

3 Proposed Approach

The goal is to compute the parameters of the polyhedral model given N aerial images. One of these images contains the external boundary of the building. We call this image the master image since it will be used as a reference image. The building boundary in the master image is provided either manually or automatically. The basic idea relies on the following fact: if the shape and the geometric parameters of the building (encoded by the vector \mathbf{w}) correspond to the real building shape and geometry, then the pixel-to-pixel mapping between the master image I_m and any other aerial image (in which the building is visible) will be correct for the entire building footprint. In other words, the dissimilarity associated with the two sets of pixels should correspond to a minimum. Recall that \mathbf{w} is defining all support planes of all the building’s facets and thus the corresponding pixel \mathbf{p}' of any pixel \mathbf{p} is estimated by a simple image transfer through homographies (3×3 matrices) based on these planes. Therefore, the associated global dissimilarity measure reaches a minimum. The global dissimilarity is given by the following score:

$$e = \sum_{j=1}^{N-1} \sum_{\mathbf{p} \in S} \rho(|I_m(\mathbf{p}) - I_j(\mathbf{p}')|) \tag{3}$$

where N is the number of aerial images in which the whole building roof is visible (in practice, N is between 2 and 5), S is the footprint of the building in the master image I_m , \mathbf{p}' is the pixel in the image $I_j \neq I_m$ that corresponds to the pixel $\mathbf{p} \in I_m$, and $\rho(x)$ is a robust error function.

The choice of the error function $\rho(x)$ will determine the nature of the global error (3) which can be the Sum of Squared Differences (SSD) ($\rho(x) = \frac{1}{2} x^2$), the Sum of Absolute Differences (SAD) ($\rho(x) = x$), or the saturated Sum of Absolute Differences. In general, the function $\rho(x)$ could be any M-estimator [9]. In our experiments, we used the SAD score since it is somewhat robust and its computation is fast.

We seek the polyhedral model $\mathbf{w}^* = (\lambda_M^*, \lambda_N^*, Z_A^*, Z_M^*, Z_N^*, Z_C^*)^T$ that minimizes the above dissimilarity measure over the building footprint:

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{arg\,min}} e \tag{4}$$

We can also measure the fitness of the 3D model by measuring the gradient norms along the projected 3D segments of the generated 3D models. In general, at facets discontinuities the image gradient is high. Thus, for a good fit, the projection of the 3D segments will coincide with pixels having a high gradient norm in all images. Therefore, we want to maximize the sum of gradient norms along these segments over all images. Recall that we have at most nine segments for our simple 3D polyhedral model. Thus, the gradient score is given by:

$$g = \frac{1}{N} \sum_j^N g_j \tag{5}$$

where g_j is the gradient score for image I_j . It is given by the average of the gradient norm over all pixels coinciding with the projected 3D model segments.

Since we want the dissimilarity measure (3) and the gradient score (5) to help us determine the best 3D polyhedral model, we must combine them in some way. One obvious way is to minimize the ratio:

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} \frac{e}{g} \quad (6)$$

It is worth noting that during the optimization of (6) there is no feature extraction nor matching among the images. The use of the image gradient norms in (6) is not equivalent to a feature-based method. In order to minimize (6) over \mathbf{w} , we use the Differential Evolution algorithm [10]. This is carried out using generations of solutions—populations. The population of the first generation is randomly chosen around a rough solution. The rough solution will thus define a given distribution for the model parameters. The rough solution is simply given by a zero-order approximation model (flat roof model) which is also obtained by minimizing the dissimilarity score over one unknown (the average height of the roof). We use the Differential Evolution optimizer since it has three interesting properties: (i) it does not need an accurate initialization, (ii) it does not need the computation of partial derivatives of the cost function, and (iii) theoretically it can provide the global optimum.

In brief, the proposed approach proceeds as follows. First, the algorithm decides if the building contains one or more facets, that is, it selects either the model of Figure 2(a) or the model of Figure 2(b). This decision is carried out by analyzing the 3D normals associated with four virtual triangles forming a partition of the whole building footprint. Second, once the model is selected, its parameters are then estimated by minimizing the corresponding dissimilarity score. Note that in the case of one facet building we only need to estimate the plane equation using the criterion (4).

4 Experimental Results

4.1 Semi-synthetic Data

We have used a real triangular roof facet in two different aerial images. The 3D shape of this facet is computed using a high resolution Digital Elevation Model. The rawbrightness of this facet is reconstructed in the second image by warping its texture in the first image using the relative geometry and the estimated 3D shape of the facet. The two textures are then perturbed by an additive uniform noise. For every noise level we run our proposed approach 10 times. For every run, we compute the error as being the difference between the estimated parameters and their ground truth values. Figure 4(a) and 4(b)

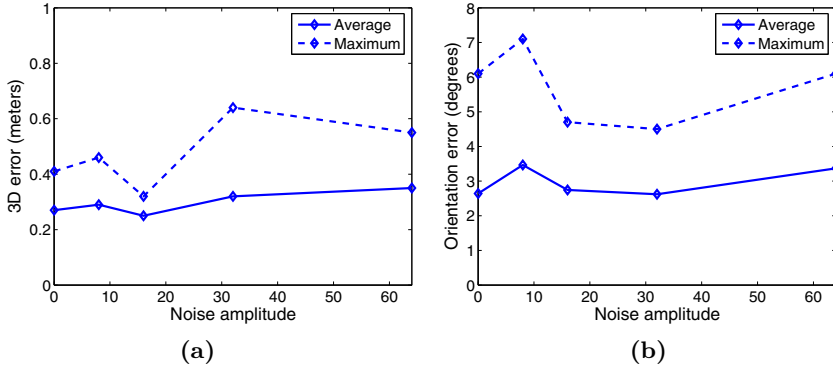


Fig. 4. 3D errors (a) and orientation errors (b) as a function of the noise amplitude

show the 3D errors (discrepancy between reconstructed and ground-truth 3D vertices) and the facet orientation error, respectively as a function of the noise amplitude.

4.2 Real Data

The proposed approach has been tested with a set of calibrated aerial images depicting a part of the city of Marseille. These data are provided by the French National Geographical Institute (IGN). The resolution of these aerial images is 4158×4160 pixels. The ratio between the baseline to the camera height is about 0.18. One pixel corresponds to a 10 cm square at ground level. Figure 5 illustrates the best model obtained at different iterations of the Differential Evolution algorithm. The projection of the model onto the first and second images is shown in the first and second columns, respectively. The third column illustrates the associated 3D model. Figure 6 illustrates the estimated model in cases where buildings are affected by shadows. Despite the presence of significant shadows the estimated polyhedral models are correct.

4.3 Method Comparison

To get quantitative evaluation we compared our method with a 3D reconstruction obtained from Digital Elevation Models (DEMs)¹. Table 1 depicts the 3D reconstruction results associated with one polyhedral model (only the heights of three vertices are shown). The first column corresponds to the reconstruction obtained with a DEM (robust plane fitting), the second column to our approach adopting the SSD function, and the third column to our approach adopting the SAD function. The last row shows the average deviation between the estimated model and the model obtained with the DEM.

¹ Although the DEMs are not ground-truth 3D data, we compared our results with them for validation purposes.

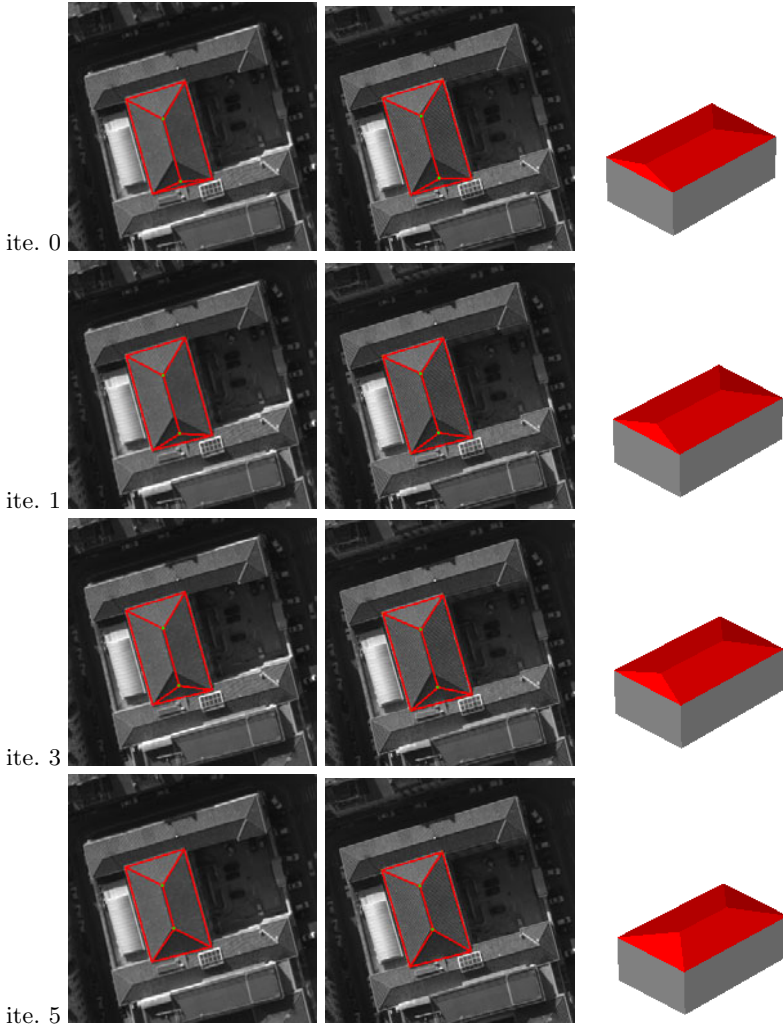


Fig. 5. The best solution at several iterations of the Differential Evolution algorithm

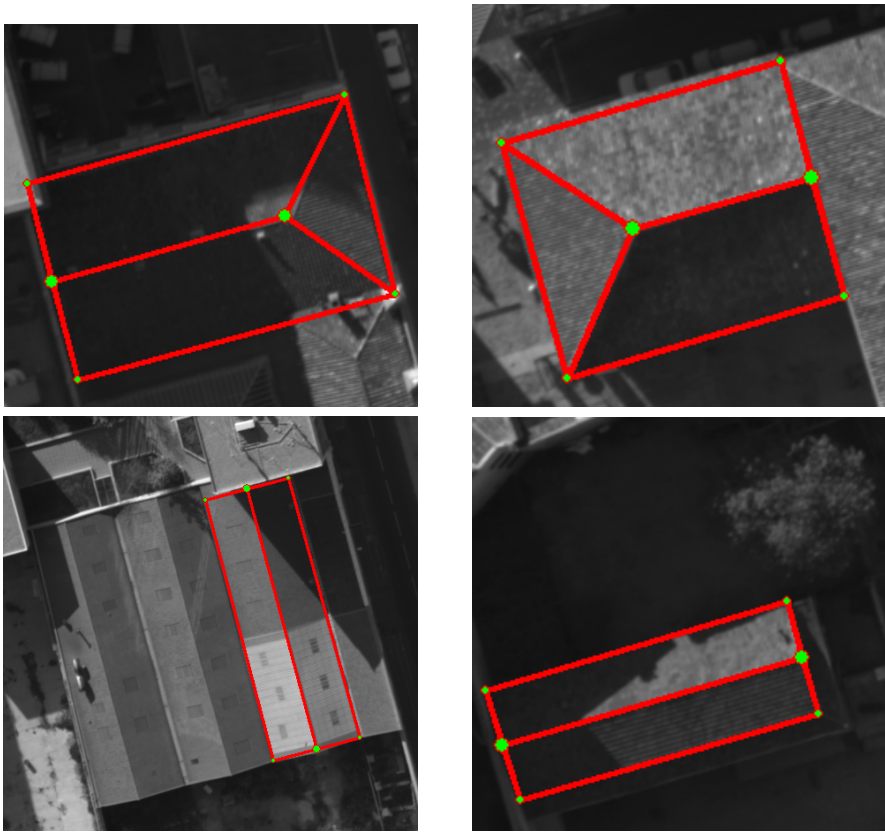


Fig. 6. Estimated 3D polyhedral models from aerial images. Despite the presence of significant shadows the approach has provided the correct models.

Table 1. Method comparison associated with one facet having three vertices. The first column depicts the estimated height of the model vertices obtained with a DEM. The second (third) column displays the estimated heights using our approach with SSD function (SAD function).

	DEM	SSD	SAD
Vertex1 height	41.96m	42.75m	42.22m
Vertex2 height	41.36m	40.87m	40.98m
Vertex3 height	39.78m	40.10m	40.22m
Average deviation	0.0m	0.53m	0.36m

5 Conclusion

We presented a direct model driven approach for extracting 3D polyhedral building models from calibrated aerial images. Unlike existing approaches, our proposal does not require feature extraction and matching in the images. Moreover, it does not rely on Digital Elevation Models. Experimental results show the feasibility and robustness of the proposed approach. Future work may investigate the extension of the approach to generic building models.

References

1. Zebedin, L., Bauer, J., Karner, K., Bischof, H.: Fusion of feature- and area-based information for urban buildings modeling from aerial imagery. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 873–886. Springer, Heidelberg (2008)
2. Brunn, A., Gulch, E., Lang, F., Forstner, W.: A multi layer strategy for 3D building acquisition. In: IAPR TC-7 Workshop: Mapping Buildings, Roads and other Man-Made Structures from Images (1996)
3. Jibrini, H., Paparoditis, N., Pierrot-Deseilligny, M., Maitre, H.: Automatic building reconstruction from very high resolution aerial stereopairs using cadastral ground plans. In: XIXth ISPRS Congress (2000)
4. Lin, C., Nevatia, R.: Building detection and description from a single intensity image. *Computer Vision and Image Understanding* 72(2), 101–121 (1998)
5. Krishnamachari, S., Chellappa, R.: Delineating buildings by grouping lines with MRFs. *IEEE Trans. on Image Processing* 5(1), 164–168 (1996)
6. Romero, T., Calderón, F.: A Tutorial on Parametric Image Registration. In: Scene Reconstruction, Pose Estimation and Tracking I-Tech (2007)
7. Taillandier, F., Vallet, B.: Fitting constrained 3D models in multiple aerial images. In: British Machine Vision Conference (2005)
8. Lafarge, F., Descombes, X., Zerubia, J., Pierrot-Deseilligny, M.: 3D city modeling based on Hidden Markov Model. In: IEEE International Conference in Image Processing (2007)
9. Chen, J., Chen, C., Chen, Y.: Fast algorithm for robust template matching with M-estimators. *IEEE Trans. on Signal Processing* 51(1), 230–243 (2003)
10. Storn, R., Price, K.: Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* 11, 341–359 (1997)

Content-Based Retrieval of Aurora Images Based on the Hierarchical Representation

Soo K. Kim¹ and Heggere S. Ranganath²

¹Clarion University of Pennsylvania
Clarion, PA 16214, USA
skim@clarion.edu

²University of Alabama in Huntsville
Huntsville, AL 35899, USA
ranganat@cs.uah.edu

Abstract. The boundary based image segmentation and representation system takes a thinned edge image and produces a unique hierarchical representation using a graph/tree data structure. The feature extraction algorithms have been developed to obtain geometric features by directly processing the graph/tree hierarchical data structure for diverse image processing and interpretation applications. This paper describes a content-based image retrieval system for the retrieval of aurora images utilizing the graph/tree hierarchical representation and the associated geometric feature extraction algorithms which extract features for the purpose of classification. The experimental results which prove that the hierarchical representation supports the fast and reliable computation of several geometric features which are useful for content based image retrieval are presented.

1 Introduction

The content based retrieval of aurora images is a subject of great interest to space scientists. Aurora is a significant phenomenon in the polar region of the Earth [1, 2]. It is a result of interaction between the solar wind and the Earth's magnetic field. Auroral events are monitored on the global scale at the Far Ultraviolet (FUV) spectrum by the Ultraviolet Imager (UVI) on board the POLAR satellite. Detecting an oval boundary of aurora is not a trivial problem because the distinction between aurora and background is not clear in most cases. In addition, the existence of dayglow emission significantly limits the automatic determination of the location and the size of auroral ovals. Auroral morphological parameters include the location and shape of the boundaries, the size of auroral ovals, and the evolution of intensified aurora arc regions during substorm events. The shape of aurora is dynamic and changes depending on the factors such as the date, time, the satellite position, etc.

Three specific types of aurora images that are of great interest to scientists have been identified. The first type, *Type1*, is defined as the aurora that has the very high magnetic latitude activity, called transpolar arc, close to the pole [1]. An example of the *Type1* aurora is shown in Fig. 1 (a). The second type, *Type2*, is defined as the aurora that is thick. In this aurora, a strong magnetic storm, often referred to as a substorm, might be present. A substorm starts as a bright spot on the auroral oval. The

spot moves in time along the oval and the intensity of the oval increases significantly around this region. Typically, a substorm is characterized by a bulge or a very thick and bright section of the oval. Space scientists are interested in studying the morphology of the substorm from birth to death. A *Type2* aurora image is shown in Fig. 1 (b). Finally, there are aurora images in which the night side is visible completely and most or all of the day side is not visible. These images are referred to as *Type3* aurora images. A *Type3* aurora is shown in Fig. 1 (c). Fig. 1 (d) shows an aurora image free of the transpolar arc and substorm, and both day and night sides are visible. Such an image is called a standard aurora image.

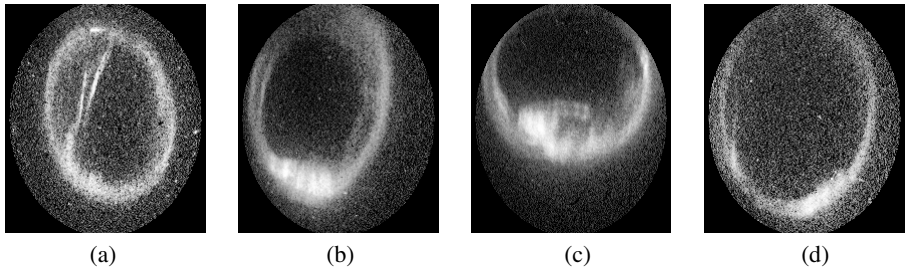


Fig. 1. Various types of aurora: (a) *Type1* aurora, (b) *Type2* aurora, (c) *Type3* aurora, and (d) Standard aurora

The process of obtaining the hierarchical representation for aurora images is described in Section 2. The geometric feature extraction algorithms that utilize the graph/tree hierarchical structure are introduced in Section 3. The process of extracting various features of aurora for the purpose of classification is discussed in Section 4. These features can be used as metadata. Section 4 also presents a method to identify the different types of aurora using the features obtained. Section 5 discusses the simulation results. Finally, the conclusion is given in Section 6.

2 Hierarchical Representation of Aurora Images

The input aurora image is thresholded to create a binary image. Although several thresholding techniques are available, for illustration purposes, the global thresholding method was used. The resulting binary image is processed using a 3x3 median filter to remove stray pixels and then an edge image is produced. This process is illustrated in Fig. 2.

The edge image obtained from the filtered binary image is processed using the boundary based image segmentation and representation system developed by Nabors [3] to create the graph/tree hierarchical representation. The image segmentation and representation system consists of four curve segment extraction networks, the line detector based on two state machines, and post processing algorithms. The curve segment extraction networks and the line detector are described in Section 2.1 and 2.2, respectively. The system receives thinned edge image as an input and produces a polyline for an open curve and a polygon for a closed curve. This information is stored in a hierarchical representation which uses both graph and tree data structures as described in Section 2.3.

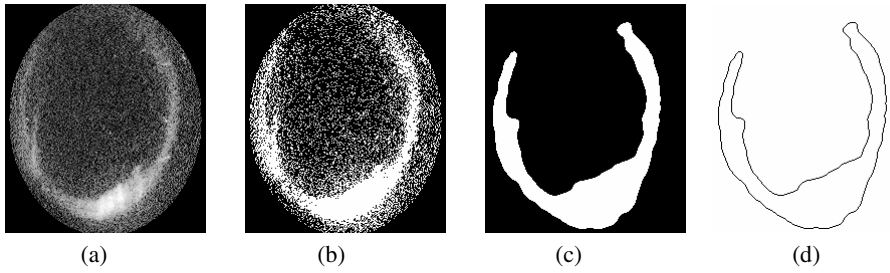


Fig. 2. Process of obtaining the edge image of an aurora: (a) Input aurora image, (b) Image after thresholding, (c) Image after median filtering, and (d) Edge image

2.1 Curve Segment Extraction Networks

The segmentation and representation system uses four curve extraction networks [3] denoted by N_1 , N_2 , N_3 , and N_4 to detect all instances of various curve segments in a binary edge image. N_1 is capable of extracting the curve segments for which the slope along the curve is in the range of $[-\infty, -1]$. Similarly, N_2 , N_3 , and N_4 detect the curve segments for which the slopes are in the range of $[-1, 0]$, $[0, 1]$, and $[1, \infty]$, respectively. The network outputs a data packet for each curve segment detected. The packet consists of the starting point of the curve segment, the number of pixels along the curve, and a binary string which encodes the curve segment using 1-bit chain code (which encodes diagonal direction using 1 and non-diagonal direction using 0).

2.2 Line Detector

The line detector [3] using two state machines M_1 and M_2 partitions a curve segment produced by the curve extraction network into line segments. The straight line characteristics are used to partition each curve segment into line segments. The first state machine M_1 receives a binary string for a curve segment from a curve extraction network. It breaks the input binary string into several disjoint parts such that each part consists of strings of zeros separated by single ones or strings of ones separated by single zeros depending on the slope of the curve. And, it counts the number of zeros separated by single ones or the number of ones separated by single zeros. The resulting string of counts is called the characteristic sequence of the curve segment. The second state machine M_2 processes the characteristic sequence produced by M_1 to partition the curve segment into straight line segments.

2.3 Hierarchical Graph/Tree Representation

The output of the segmentation system is represented by a hierarchical graph/tree data structure [3]. Each curve is represented by a node in the graph in which an edge indicates that the two corresponding curves are connected. Each node of the graph is the root node of the tree data structure that represents the corresponding curve. Each curve segment of the curve extracted by a curve extraction network is represented by a node on the first level (Level-1 node) of the tree and nodes on the second level (Level-2 nodes) of the tree identify the line segments into which a curve segment is divided by the state machines.

3 Feature Extraction

We have developed computationally efficient algorithms [4] to extract general shape features of a curve, and the convex hull and the minimum bounding rectangle of a closed curve. These geometric features are obtained directly from the graph/tree hierarchical data structure that contains the boundary information produced by the segmentation system. As the hierarchical representation permits fast computation of several features of open and closed curves, the approach is referred to as the *transform-and-conquer approach*.

Section 3.1 illustrates the terminology used in the feature extraction algorithms. Section 3.2 discusses the general shape features of a curve such as concave-up, concave-down, local minimum, local maximum, inflection points, and concavities. Section 3.3 and Section 3.4 briefly introduce the methods for finding the convex hull and the minimum bounding rectangle, respectively. The details can be found in [4].

3.1 Terminology

The *network sequence* of a curve is defined as the ordered sequence of the curve extraction network numbers which extract the curve segments of the curve. The magnitude of the i^{th} element of the *network difference sequence* is obtained as the absolute difference of the $(i+1)^{\text{th}}$ and i^{th} elements of the *network sequence*. The sign of the i^{th} element of *network difference sequence* is determined based on the orientation of the $(i+1)^{\text{th}}$ curve segment relative to the i^{th} curve segment. If the $(i+1)^{\text{th}}$ curve segment lies to the right of the i^{th} curve segment, then the sign is negative. Otherwise, it is positive. The *slope differential sequence* is obtained by adding contiguous blocks of elements of the same sign in the *network difference sequence*.

3.2 General Shape Features

The general shape attributes of a curve such as concave-up, concave-down, local minimum, local maximum, inflection points, and concavities can be identified by simply using the *network sequence*, the *network difference sequence*, and the *slope differential sequence* of the curve.

Concave-Up and Concave-Down. An open curve is concave-up over an interval if the first derivative is increasing over the interval. Therefore, a *network sequence* of increasing numbers identifies concave-up portion of the curve. Similarly, a *network sequence* of decreasing numbers identifies concave-down portion of the curve. Also, the positive and negative numbers in the *slope differential sequence* identify concave-up and concave-down segments of the curve, respectively. The *network sequence* and the *slope differential sequence* for the curve in Fig. 3 are [4 3 2 1 2 3 4 3 2 1] and [-3 +3 -3], respectively. Note that the arrow indicates the starting point of the curve and the curve is traversed from left to right. From these sequences, one can easily determine the general shape of the curve which consists of two concave-down segments and one concave-up segment.

Local Minimum and Local Maximum. The location of each local minimum of a curve is identified by a transition from the curve extraction network N_1 or N_2 to the

curve extraction network N_3 or N_4 . This is because the curve segment extracted by N_1 or N_2 has a negative slope and the curve segment extracted by N_3 or N_4 has a positive slope. A transition from a curve with a negative slope to a curve with a positive slope defines a local minimum. Similarly, a local maximum is identified by a transition from the curve extraction network N_3 or N_4 to the curve extraction network N_1 or N_2 . This is because the curve segment extracted by N_3 or N_4 has a positive slope and the curve segment extracted by N_1 or N_2 has a negative slope. A transition from a curve with a positive slope to a curve with a negative slope defines a local maximum. Therefore, it is obvious that the curve in Fig. 3 has two local maxima and one local minimum, and their locations are also known.

Inflection Point and Concavity. An inflection point is defined as a point where the curve changes from concave-up to concave-down or vice versa. Each sign change in the *slope differential sequence* identifies an inflection point. In general, the number of inflection points is equal to the length of the *slope differential sequence* for a closed curve and (length of the *slope differential sequence* - 1) for an open curve. Therefore, the curve in Fig. 3 has two inflection points as identified by two sign changes in its *slope differential sequence* [-3 +3 -3]. The first inflection point is on the curve segment extracted by N_1 (fourth curve segment) and the second inflection point is on the curve segment extracted by N_4 (seventh curve segment). For a closed curve, the number of inflection points is even. The inflection points are useful in identifying the number and location of concavities. The convex hull algorithm presented in the next section makes use of inflection points to identify concavities.

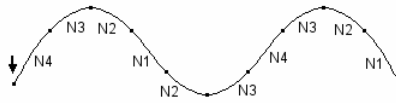


Fig. 3. An open curve that consists of ten curve segments

3.3 Method for Finding the Convex Hull

The new convex hull algorithm utilizes the hierarchical representation of the concave object and produces the modified hierarchical representation for the resulting convex polygon. The algorithm uses a two-step approach. While Step 1 identifies the concavities at the curve segment level, Step 2 identifies the concavities missed in Step 1 using line segments. Step 1 is based on the observation that a positive number in the *network difference sequence* indicates the presence of a concavity and also provides rough information of its location. The algorithm uses a simple decision function [5] to determine if a given point lies to the left or right side of a given line. Let $P_0(x_0, y_0)$, $P_1(x_1, y_1)$, and $P_2(x_2, y_2)$ be the three points. Using two vectors $P_1 - P_0$ and $P_2 - P_0$, the value of the decision function D is obtained by the following equation:

$$D = [(x_1 - x_0)(y_2 - y_0) - (x_2 - x_0)(y_1 - y_0)]. \quad (1)$$

It has been shown that D is positive if P_2 is to the left of P_0P_1 , and D is negative if P_2 is to the right of P_0P_1 . If all three points are collinear, then D is zero.

Fig. 4 (a) shows a concave object. For this object, Step 1 takes 27 iterations and Fig. 4 (b) through Fig. 4 (e) show the intermediate results after four selected iterations (iteration 3, 9, 15, and 21). Fig. 4 (f) is the result of Step 1 after 27 iterations and Fig. 4 (g) is the result of Step 2 after 3 iterations. The resulting convex hull of the object is shown in Fig. 4 (h).

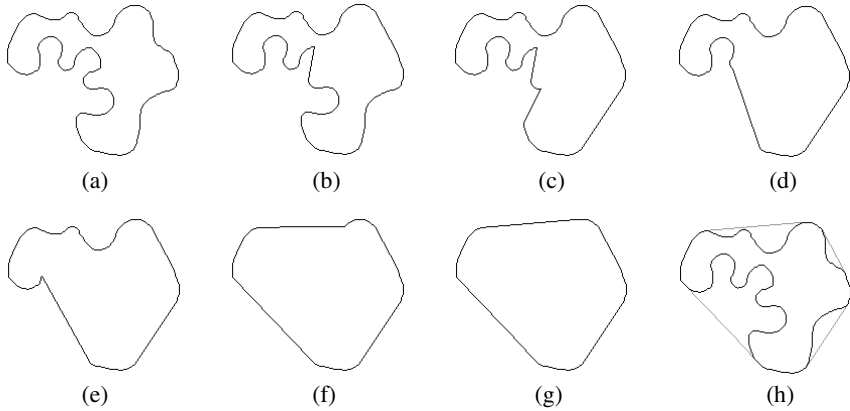


Fig. 4. Illustration of Step 1 and Step 2 of the convex hull algorithm

3.4 Method for Finding the Minimum Bounding Rectangle

The new algorithm for finding the minimum bounding rectangle is based on two theorems proven by Freeman and Shapira [6]. The four steps of our algorithm are given below.

Theorem 1. The rectangle of minimum area enclosing a convex polygon has a side collinear with one of the edges of the polygon.

Theorem 2. The minimum-area rectangle encasing the convex hull of a simple, closed, chain-coded curve is one and the same as the minimum-area rectangle encasing the curve.

Step 1. For a convex object, this step is skipped. For a concave object, the convex hull of the given object and its hierarchical representation are obtained using the algorithm described in Section 3.3. The straight line segments of the convex hull specified by the Level-2 nodes represent the object as a convex polygon. A concave object and its convex polygon are shown in Fig. 5 (a) and Fig. 5 (b).

Step 2. From *Theorem 1* and *Theorem 2* described above, the minimum bounding rectangle must have an edge collinear with one of the edges of the polygon. The construction of the bounding rectangle that is collinear with a hull edge is illustrated by constructing the bounding rectangle that is collinear with hull edge AB in Fig. 5 (c). Let m be the slope of the line AB . Let p be the starting point of a curve segment that is farthest from the line AB . Let $p1$ be the point that is farthest from AB which is obtained by trial-and-error starting from p .

Step 3. Two points p_2 and p_3 are found such that the distance between the lines with slope $(-1/m)$ passing through p_2 and p_3 is maximum. These points can be easily determined by the method used to find p_1 . After finding p_1 , p_2 , and p_3 , the bounding rectangle is determined by calculating the four corner points q_1 , q_2 , q_3 , and q_4 .

Step 4. Step 2 and Step 3 are repeated for each edge (line segment) of the convex polygon, and the bounding rectangle that has the minimum area is selected. For the object in Fig. 5 (a), the bounding rectangle corresponding to the hull edge AB happens to be the minimum bounding rectangle which is shown in Fig. 5 (d).

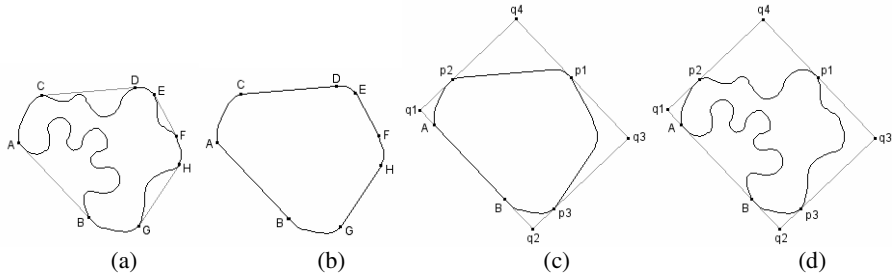


Fig. 5. Illustration of the minimum bounding rectangle algorithm: (a) A concave object and its convex hull with four hull edges AB , CD , EF , and GH , (b) The convex polygon of the object in (a) constructed using four hull edges, (c) The bounding rectangle formed using the hull edge AB , and (d) The resulting minimum bounding rectangle

4 Method for Identifying the Types of Aurora

In order to build a content-based image retrieval system for aurora images, one must identify the list of features which can be extracted from the hierarchical representation and are able to classify images into the desired categories. The extent of the aurora oval along the two coordinate axes, the orientation of the oval, the presence or absence of the transpolar arc, the orientation of the transpolar arc, the circularity measure, the percent of the oval that is visible, the maximum thickness of the oval, and the location of the aurora itself have been selected. These features appear to be adequate for identifying *Type1*, *Type2*, and *Type3* aurora images. Section 4.1, 4.2, and 4.3 discuss the determination processes of *Type1* aurora, *Type2* aurora, and *Type3* aurora, respectively. It is shown that all the above features can be computed from the hierarchical representation.

4.1 Determination of *Type1* Aurora

A *Type1* aurora is characterized by the transpolar arc as shown in Fig. 6 (a). The edge image which is used to create the hierarchical representation is also shown in Fig. 6 (a). The hierarchical representation consists of two graph nodes that are not connected to each other. If a transpolar arc is present, then the inner closed curve will have at least one deep concavity. The location and depth of concavities, if present, can be determined from the hierarchical representation using the method described in Section 3.

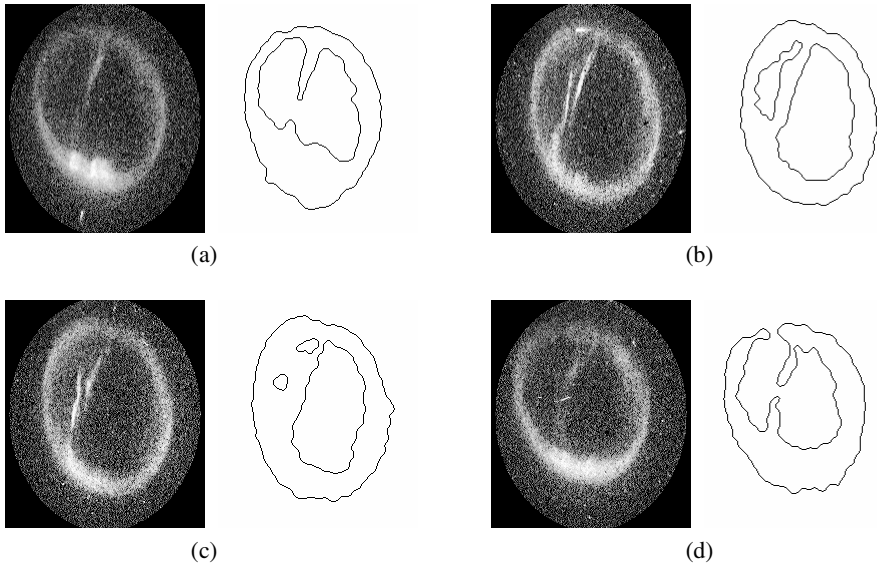


Fig. 6. Four *Type 1* aurora images with transpolar arcs and their edge images

Aurora images with transpolar arcs come in many forms. Three additional aurora images with transpolar arcs and their edge images which have different hierarchical representations are shown in Fig. 6 (b), Fig. 6 (c), and Fig. 6 (d). For the image in Fig. 6 (b), the edge image has three closed curves (3 graph nodes, no graph edges) with two closed curves completely inside the larger outer curve. Similarly, the edge image in Fig. 6 (c) consists of four closed curves (4 graph nodes, no graph edges) with three curves located completely within the outer closed curve. In fact, the images in Fig. 6 (b) and Fig. 6 (c) are almost identical. The difference in their edge images is due to poor segmentation which is not very uncommon and should be expected. Therefore, two or more curves nested inside an outer curve are also considered as an indication of the presence of a transpolar arc. Finally, the image may map to one complex closed curve as shown in Fig. 6 (d). In this case, the presence of a transpolar arc is also detected simply by finding deep concavities. The average of all the slopes of the line segments forming the concavity can be used as an approximation to the orientation of the transpolar arc.

4.2 Determination of *Type 2* Aurora

Aurora images with thick aurora are of *Type 2*. They could contain substorms. The thickness of the aurora oval is useful in recognizing substorms. The method used for the determination of the thickness of the oval is given below.

Step 1. The approximate location of the centroid C of the aurora oval is computed by averaging the coordinates of the starting points of all the line segments.

Step 2. In order to determine the thickness associated with the starting point P of a line segment on the outer boundary of the aurora, the point Q at which line PC intersects the inner boundary is found. The distance between P and Q is taken as the thickness of the aurora oval at P.

Step 3. The values of the thickness associated with the starting points of all line segments on the outer boundary are determined by repeating Step 2. If the maximum thickness is greater than a predetermined threshold, then the image is taken as a *Type2* aurora image.

The outer boundary of the edge image of the aurora in Fig. 7 (a) consists of 50 line segments. The centroid of the oval and all 50 radial lines are shown in Fig.7 (b). The plot in Fig. 7 (c) shows the thickness of the aurora oval along the boundary.

A *Type2* aurora is an aurora in which a magnetic substorm is possibly present. The maximum thickness of the oval begins to increase as the substorm begins and intensifies. The maximum thickness of the oval decreases as the substorm subsides. Therefore, a sequence of *Type2* aurora images could be identified as a substorm by tracking the maximum thickness of the oval from frame to frame. From the locations at which the thickness peaks in the oval thickness plots, the location and the movement of the storm are traced.

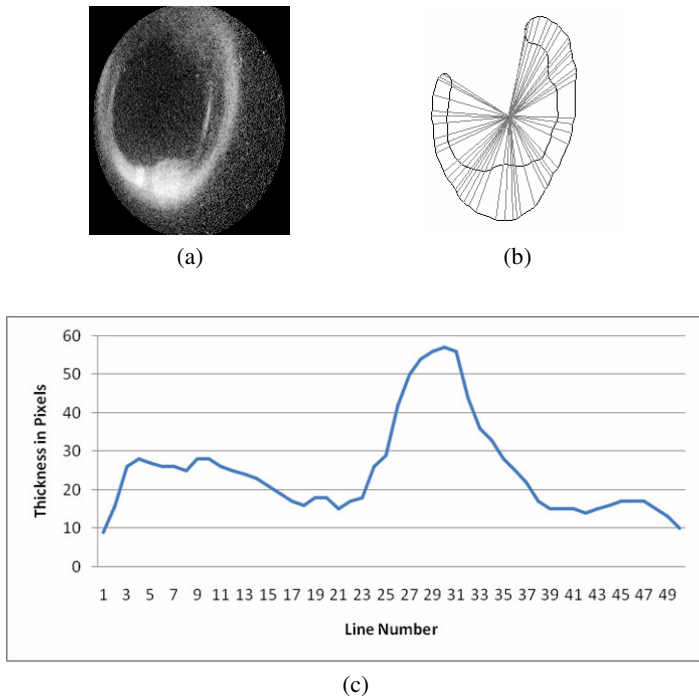


Fig. 7. Illustration of the computation of the aurora oval thickness for *Type2* aurora

4.3 Determination of *Type3* Aurora

In a *Type3* aurora image, the night side of the aurora is visible and most of the day side of the aurora is not visible. The edge image of a typical *Type3* aurora usually consists of one closed curve with a large and wide concavity as shown in Fig. 8 (b). The method used to identify a *Type3* aurora is given below.

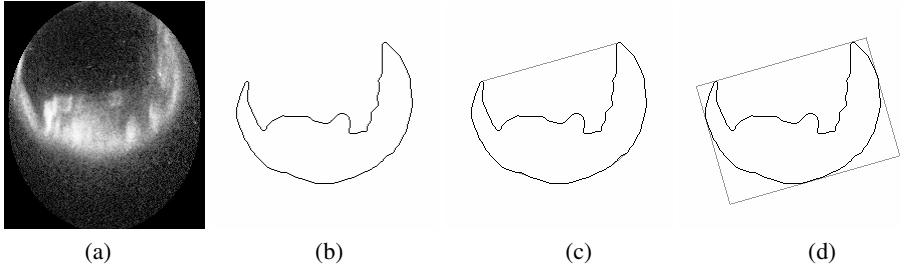


Fig. 8. Illustration of the determination of *Type3* aurora: (a) Aurora image, (b) Edge image, (c) Convex hull of the edge image, and (d) Minimum bounding rectangle

Step 1. The length of the convex hull edge that bridges the widest concavity is determined. The hull edge for the edge image in Fig. 8 (b) is given in Fig.8 (c).

Step 2. The extent of the aurora oval in a direction of the hull edge determined in Step 1 is found. The algorithm that finds the minimum bounding rectangle given in Section 3.4 is used for this purpose. This is illustrated in Fig.8 (d).

Step 3. If the ratio of the length of the hull edge in Step 1 to the extent determined in Step 2 is greater than a predetermined threshold, then the image is taken as a *Type3* aurora image.

5 Simulation Results

Precision and recall [7] are used in a content-based retrieval system to measure the performance of the retrieval. Precision is the ratio of the number of the relevant images retrieved (Nr) to the total number of images retrieved (Nt). Recall is the ratio of the number of the relevant images retrieved (Nr) to the total number of relevant images in the database (Nd).

Forty sample aurora images, ten for each type of aurora including the standard aurora, are analyzed to obtain the parameters for the decision of types. For *Type1* aurora, the ratio of the depth of the concavity to the length of the side of the minimum bounding rectangle of the inner boundary that is roughly oriented in the direction of the concavity is found to be greater than 0.35. For *Type2* aurora, the value of the thickness measured in pixels is found to be greater than 45. For *Type3* aurora, the ratio of the length of the hull edge that bridges the widest concavity to the extent of the minimum bounding rectangle in the direction of the hull edge is found to be greater than 0.65.

A database of 129 images, which does not include the 40 sample images used to derive the threshold values of the parameters for the classification purpose, was created. There are 45 *Type1*, 27 *Type2*, and 37 *Type3* aurora images. Some of the aurora images belong to two types. These images are from the time period 1997 to 1999, which are used in the experiment by Cao et al. [8] The database is searched for the automatic retrieval of each type of aurora using the approach described previously. Table 1 shows precision and recall of *Type1*, *Type2*, and *Type3* aurora retrieval.

Table 1. Precision and recall of three types of aurora

	Precision			Recall		
	Nr	Nt	Nr/Nt	Nr	Nd	Nr/Nd
<i>Type1</i>	42	42	100%	42	45	93.3%
<i>Type2</i>	23	26	88.5%	23	27	85.2%
<i>Type3</i>	37	37	100%	37	37	100%

A study of the misclassified images reveals that the misclassification was mainly due to poor preprocessing. It is possible to achieve better accuracy by improving the segmentation and preprocessing steps. In conclusion, the feasibility of building a content-based image retrieval system based on the hierarchical representation is demonstrated.

6 Conclusion

In this paper, we have presented the content-based retrieval system for aurora images. The system utilizes the graph/tree hierarchical representation obtained from the boundary based image segmentation and representation system and extracts various geometric features for the purpose of classification. Those features include general shape attributes of a curve, the convex hull, and the minimum bounding rectangle of an object. The experimental results have proven that the hierarchical representation supports the fast and reliable computation of several geometric features and those geometric features extracted directly from the hierarchical representation are useful for content based image retrieval and also for a wide range of image interpretation applications. Other applications such as shape matching under rotation and scale changes and recognition of the license plate can be found in [4].

References

1. Li, X., Ramachandran, R., Movva, S., Graves, S., Germany, G., Lyatsky, W., Tan, A.: Dayglow Removal from FUV Auroral Images. In: Proceedings of IEEE International Geoscience and Remote Sensing Symposium, vol. 6, pp. 3774–3777 (2004)
2. Li, X., Ramachandran, R., He, M., Movva, S., Rushing, J., Graves, S., Lyatsky, W., Tan, A., Germany, G.: Comparing Different Thresholding Algorithms for Segmenting Auroras. In: Proceedings of International Conference on Information Technology: Coding and Computing, vol. 2, pp. 594–601 (2004)

3. Nabors, D.H.: A Boundary Based Image Segmentation and Representation Method for Binary Images, Doctoral Dissertation. The University of Alabama in Huntsville (2000)
4. Kim, S.K.: Hierarchical Representation of Edge Images for Geometric Feature Based Image Interpretation, Doctoral Dissertation. The University of Alabama in Huntsville (2007)
5. Sunday, D.: Area of Triangles and Polygons (2D & 3D),
[http://softsurfer.com/Archive/algorithm_0101/
algorithm_0101.htm](http://softsurfer.com/Archive/algorithm_0101/algorithm_0101.htm)
6. Freeman, H., Shapira, R.: Determining the Minimum-Area Encasing Rectangle for an Arbitrary Closed Curve. *Communication of the ACM* 18, 409–413 (1975)
7. Deb, S., Zhang, Y.: An Overview of Content-Based Image Retrieval Techniques. In: *International Conference on Advanced Information Networking and Applications*, vol. 1, pp. 59–64 (2004)
8. Cao, C., Newman, T., Germany, G.: Shape-Based Mechanisms for Content-Based Retrieval of Aurora Images. In: *Proceedings of SPIE, Wavelet Applications in Industrial Processing IV*, vol. 6383, pp. 63830T-1–63830T-12 (2006)

Improved Grouping and Noise Cancellation for Automatic Lossy Compression of AVIRIS Images

Nikolay Ponomarenko¹, Vladimir Lukin¹, Mikhail Zriakhov¹, and Arto Kaarna²

¹ National Aerospace University

Department of Transmitters, Receivers and Signal Processing

17 Chkalova Street, 61070 Kharkov, Ukraine

² Lappeenranta University of Technology

Department of Information Technology

Machine Vision and Pattern Recognition Laboratory

P.O. Box 20, FI-53851 Lappeenranta, Finland

uagames@mail.ru, lukin@ai.kharkov.com, arto.kaarna@lut.fi

Abstract. An improved method for the lossy compression of the AVIRIS hyperspectral images is proposed. It is automatic and presumes blind estimation of the noise standard deviation in component images, their scaling (normalization) and grouping. A 3D DCT based coder is then applied to each group to carry out both the spectral and the spatial decorrelation of the data. To minimize distortions and provide a sufficient compression ratio, the quantization step is to be set at about 4.5. This allows removing the noise present in the original images practically without deterioration of the useful information. It is shown that for real life images the attained compression ratios can be of the order 8 ... 35.

Keywords: remote sensing, hyperspectral images, noise estimation, noise cancellation, image compression, decorrelation.

1 Introduction

Hyperspectral imaging has gained wide popularity in recent two decades [1] [2]. Remote sensing (RS) hyperspectral images (HSI) as those ones formed by the AVIRIS, HYPERION, CHRIS-PROBA and other sensors are characterized by large amount of data [1]-[3]. Thus, their compression for transferring, storage, and offering to users is desirable.

Even the best lossless coders provide a compression ratio (CR) not larger than 4 for such data [3], [4], and this is often not appropriate. Therefore, the application of the lossy compression becomes necessary [3], [5]-[8]. There are many methods of HSI lossy compression already developed. To be efficient, these methods have to exploit both the sufficient spectral (inter-channel) and spatial correlation of the data inherent for HSI [2]. This is usually done by carrying out the spectral decorrelation first which is followed by reducing the spatial redundancy.

The known methods employ various mathematical tools including the vector quantization [9], the principal component analysis [10], the orthogonal transforms [11] and combined approaches where similar transforms (like 3-D wavelets or 3-D JPEG) are used for the spectral and the spatial decorrelation of the data [12]. Whilst the independent and the principal component analysis (ICA and PCA) methods have been basically recommended for the spectral decorrelation of bands, other orthogonal (different wavelet and discrete cosine) transforms have been mainly exploited for decreasing the spatial redundancy in HSI. This is explained by the fact that the ICA and the PCA techniques are more common in classification based on spectral features, whilst DCT and wavelets are put into basis of the modern standards JPEG and JPEG2000 used for 2-D data (image) lossy compression [13], [14].

An important item in the lossy compression of HSI is to take into account the fact that the original images are noisy and the signal-to-noise ratio (SNR) is considerably different in different sub-band images [15]. Then, if losses mainly relate to the noise removal (image filtering), such lossy compression can be useful in two senses. First, the data size reduction is provided. Second, images are filtered [16] and this leads to a better classification of the decompressed HSI. Note that similar approaches have been considered in astronomy [17] and it has been demonstrated that the lossy compression under certain conditions does not lead to the degradation of object parameters measurements for the compressed images.

In general, there are two options in compressing AVIRIS and other hyperspectral data. One option is to compress the radiance data and the other variant is to apply coding to the reflectance data. Below we considered the latter approach since it has been shown in [18] that it leads to smaller degradations.

Two basic requirements are to be satisfied in the lossy compression. The statistical and spatial correlation characteristics of the noise are to be carefully taken into consideration to introduce minimal losses in the image content [19], [20]. Besides, it is desirable to carry out the compression in an automatic manner, i.e., in a non-interactive mode. Note that a provided CR depends on both the noise level (the type and the statistical characteristics) and the image content [19], [21]. The requirement to increase the CR as possible remains important as well.

Fortunately, there exist methods for the blind evaluation of the noise statistics [22], [23]. Methods that operate in the spectral domain are able to evaluate the variance of additive i.i.d. noise quite accurately even if the image is rather textural [22], [25], [27]. However, these methods produce biased estimates in cases of the spatially correlated noise. The estimates might be considerably smaller than true values of the noise variance. Note that the noise in AVIRIS images is not i.i.d. [23].

The aforementioned properties of the estimates of the noise statistics are taken into consideration in design of the modified method for the automatic lossy compression of the AVIRIS images. In fact, below we show how it is possible to improve the performance of the method earlier proposed in [16]. The positive

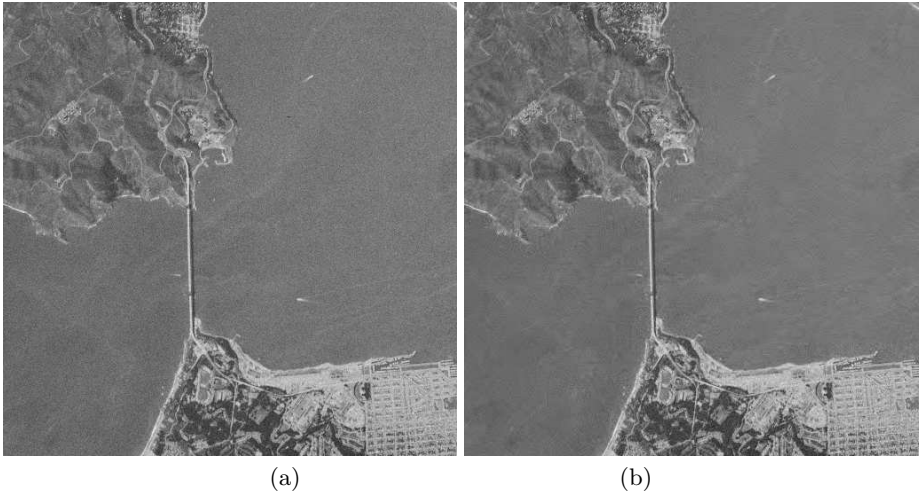


Fig. 1. Noisy (a) and compressed (b) Frisco images

effect is provided due to the specific normalization of the sub-band images before their compression and due to the larger size of the groups composed. Besides, we consider in more detail how the coder parameters are set.

2 Lossy Compression of One-Channel Image

The lossy compression of one-channel noisy images has certain peculiarities. One of them is the noise filtering effect [24]. This effect is positive in the sense of improving the image quality and enhancing its classification [25] but only under the condition that the compression ratio (coder parameters) is adjusted to the noise type and statistics. Consider a simple example. The mixed additive and signal dependent i.i.d. Gaussian noise has been added to the gray-scale test image Frisco (Fig. 1 a)) where the additive noise with the variance $\sigma^2 = 64$ was predominant (the variance of the signal dependent noise $\sigma^2 = kI_{ij}^{tr}$, $k = 0.1$, I_{ij}^{tr} denotes a true value of ij -th pixel).

The noisy image has been subject to the lossy compression by the DCT based coder AGU [26] controlled by the quantization step (QS). QS was set equal to $\beta\sigma$ with β from 0.5 to 6. Two curves have been obtained (Fig. 2), $PSNR_{or}(QS)$ and $PSNR_{nf}(QS)$ where the former is determined for the decompressed and the original (noise added) images and the latter one for the decompressed and the noise-free images. The curve $PSNR_{or}(QS)$ is monotonous. $PSNR_{or}$ decreases with the larger QS . The curve $PSNR_{nf}(QS)$ exhibits a maximum that is observed for $QS = 4.5\sigma = 36$. Two equal values of $PSNR_{nf}$ (e.g., equal to 32dB) take place for $QS_1 = 3.5\sigma = 28$ and $QS_2 = 6\sigma = 48$. In the first case, a less efficient noise suppression but a better edge-detail preservation are observed (Fig. 1 b)) and vice versa. Thus, setting $QS_2 \approx 4.5\sigma$ can be a good choice.

In practice, the standard deviation σ can be unknown in advance. Then its value $\hat{\sigma}$ should be estimated for an image to be compressed and the quanti-

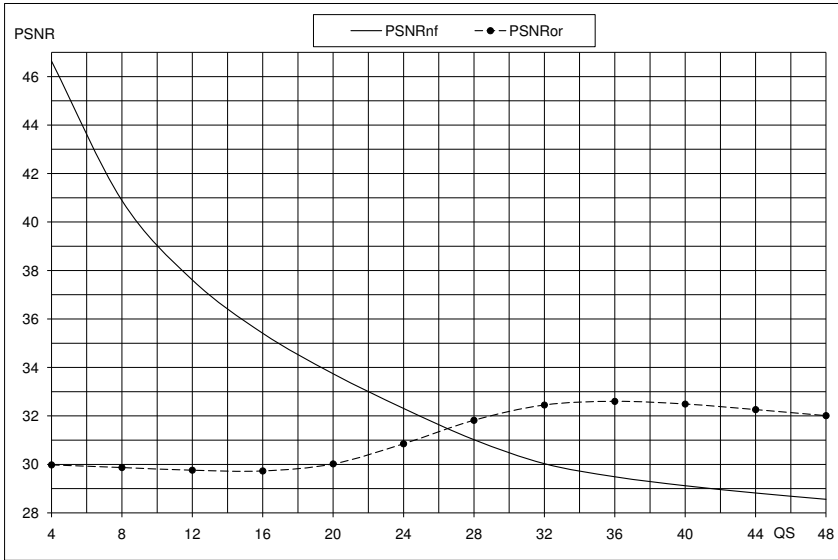


Fig. 2. $PSNR_{or}(QS)$ and $PSNR_{nf}(QS)$ for the one-channel Frisco test image

zation step is to be set as $QS = 4.5\hat{\sigma}$. Since the estimate $\hat{\sigma}$ is not absolutely accurate, some oversmoothing or undersmoothing of the noise due to the lossy compression might take place. The undersmoothing that is observed for $\hat{\sigma} < \sigma$ is less dangerous, but then the attained compression ratio is smaller than can be reached for $QS = 4.5\sigma$.

The existence of the image-dependent noise in AVIRIS-images and the noise-dependent lossy compression is demonstrated in Fig. 3. In Fig. 3, a) there is a sample of a noisy channel from the image Cuprite, band 2. Then the band is compressed/decompressed with the quantization step $QS(n) = 4.5\hat{\sigma}_n$ where n is the index for a sub-band. The reconstructed band is in Fig. 3, b). The denoising is now well observed. In Fig. 3, c) there is the band 30 from the Cuprite image having high SNR. In turn, Fig. 3, d) demonstrates the decompressed image. As seen, no distortions (losses) are observed visually. Thus, we can state that for the sub-bands with rather low SNR the proposed approach to the lossy compression provides efficient denoising whilst the useful information contained in the sub-band images characterized by a high SNR is preserved well.

AVIRIS images have different contents depending on the area under imaging. In Fig. 4 two bands from the Moffett Field image are shown. On left, there are the original images and on right, there are the compressed/reconstructed images. On top row, the band 128 is shown, on bottom row, the band 223 is shown. Compared to the Cuprite image in Fig. 3, there are more tiny details in the Moffett Field image. From the compression point of view, the more complex structures mean a lower compression ratio. From the denoising point of view, the proposed approach is operating similarly to a less detailed image: the quantization step is still determined from the noise characteristics only.

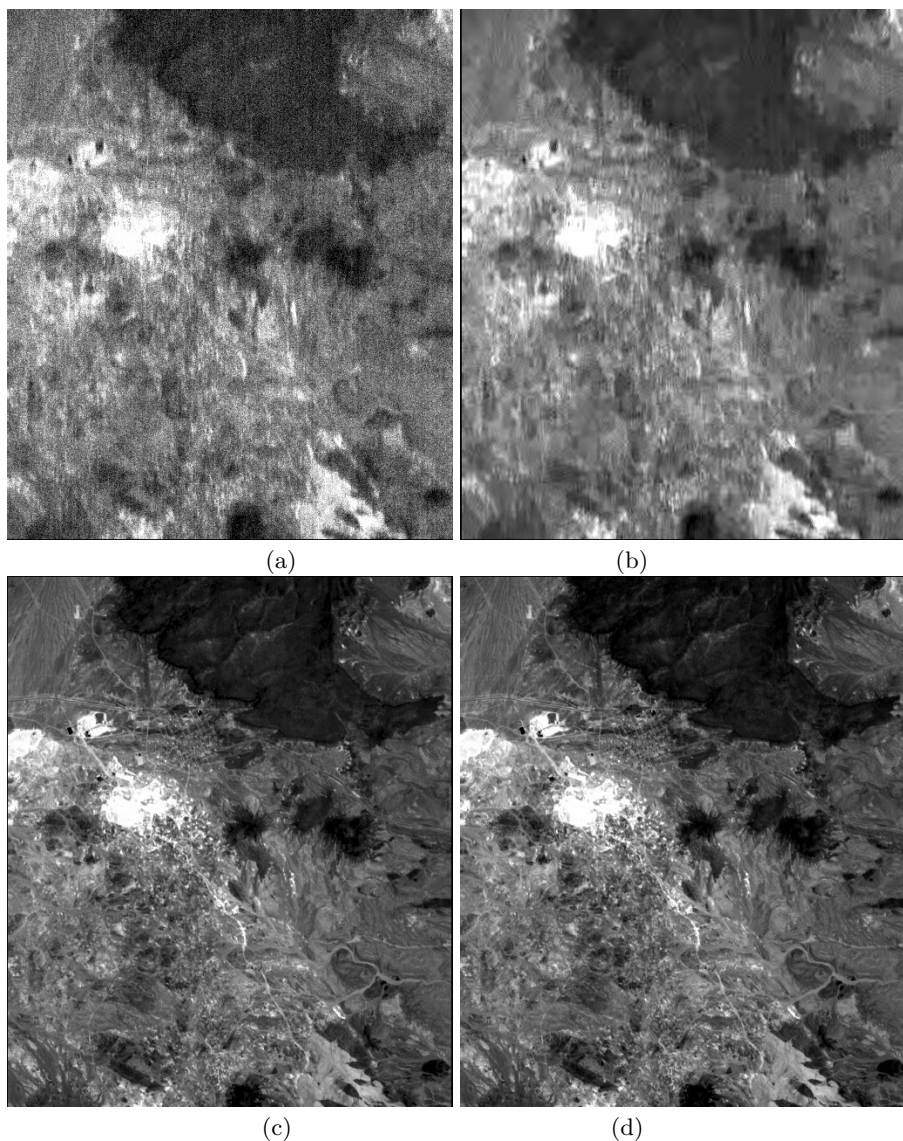


Fig. 3. The noisy sub-band image (Cuprite, channel 2): original (a) one and the image after compression/decompression (b). The sub-band image with high SNR (Cuprite, channel 30), original (c) one and the image after compression/decompression (d).

3 Improved Method for Compressing AVIRIS Data

In the earlier study [16], two methods of the HSI automatic lossy compression have been proposed. For both of them, the first stage is the blind evaluation of the additive noise variance $\hat{\sigma}_n^2$, $n = 1, \dots, 224$ (the AVIRIS imager has 224 sub-bands). The method called M1 in [16], presumes a component-wise lossy

compression of the data with setting the quantization step individually for each sub-band image as $QS(n) = 4.5\hat{\sigma}_n$. This method is simple but it in no way exploits the inter-band correlation inherent for hyperspectral data [2]-[6]. Note that the accounting for the spectral redundancy of HSI results in a considerable increase of CR. Taking this into account, a method called M2 has been also proposed in [16]. For this method, the grouping of sub-band images is to be carried out using two main rules. The first rule is that each the k -th group should contain 4, 8, or 16 sub-bands. The second rule is that the grouping is started from the first sub-bands and is performed depending upon the variation of the noise variance. The main condition checked is

$$\frac{\hat{\sigma}_{n,k \max}^2}{\hat{\sigma}_{n,k \min}^2} \leq 2 \quad (1)$$

where $n \in G_k$, $\hat{\sigma}_{n,k \max}^2$, $\hat{\sigma}_{n,k \min}^2$ are the maximal and the minimal noise variances, respectively, in a group G_k . For each k -th group, the quantization step $QS_k = 4.5\hat{\sigma}_{n,k \min}$ for compressing a given group of sub-band images. The compression is carried out by the 3D AGU coder [16] based on the discrete cosine transform that performs both the spectral and the spatial decorrelation of the data. The smallest $\hat{\sigma}_{n,k \min}$ in a group is used for calculating QS_k to avoid the oversmoothing of the compressed images (see Section 2). Note that the method M2 produces about twice larger CR than the method M1 with smaller introduced distortions. Fig. 5 presents an example of the estimated noise standard deviations $\hat{\sigma}_n$, $n = 1, \dots, 224$ and the set quantization steps for the methods M1 and M2. As it is seen, the group sizes for the method M2 are different and they are small (4 sub-bands) for subsequent sub-bands with a high variation of $\hat{\sigma}_n$.

The method M2 described in [16] has a certain shortcoming. If a group size is small (e.g., 4 sub-bands), this does not allow exploiting the spectral redundancy in full extent (note that the spectral decorrelation in many modern coders in HSI is carried out for all sub-bands [5] although such approach might lead to undesirable effects [28]). The reason why for some groups their size is small is the use of the condition given in Eq. 1. However, there is a quite simple opportunity to overcome the limitations on the group size as well as the problems of the variation of the noise variance and the sub-band image undersmoothing.

The idea is to make all $\hat{\sigma}_n^2$, $n \in G_k$ equal to each other before the compression. This can be easily done by the following normalization:

$$I_{ij,n}^{norm} = \frac{I_{ij,n}}{\hat{\sigma}_n}, \quad n = 1, \dots, 224 \quad (2)$$

where $I_{ij,n}$ is an original image value at ij -th pixel of n -th sub-band. Such normalization allows providing the additive noise variance in all images close to the unity (with taking into account the accuracy of the blind estimation).

After the normalization given in Eq. 2, the lossy compression is applied to the sub-band images collected into groups with the size $Q > 4$. For all these groups, QS is the same and, since the standard deviation of the additive noise in all images after the normalization becomes about 1.0, we recommend using

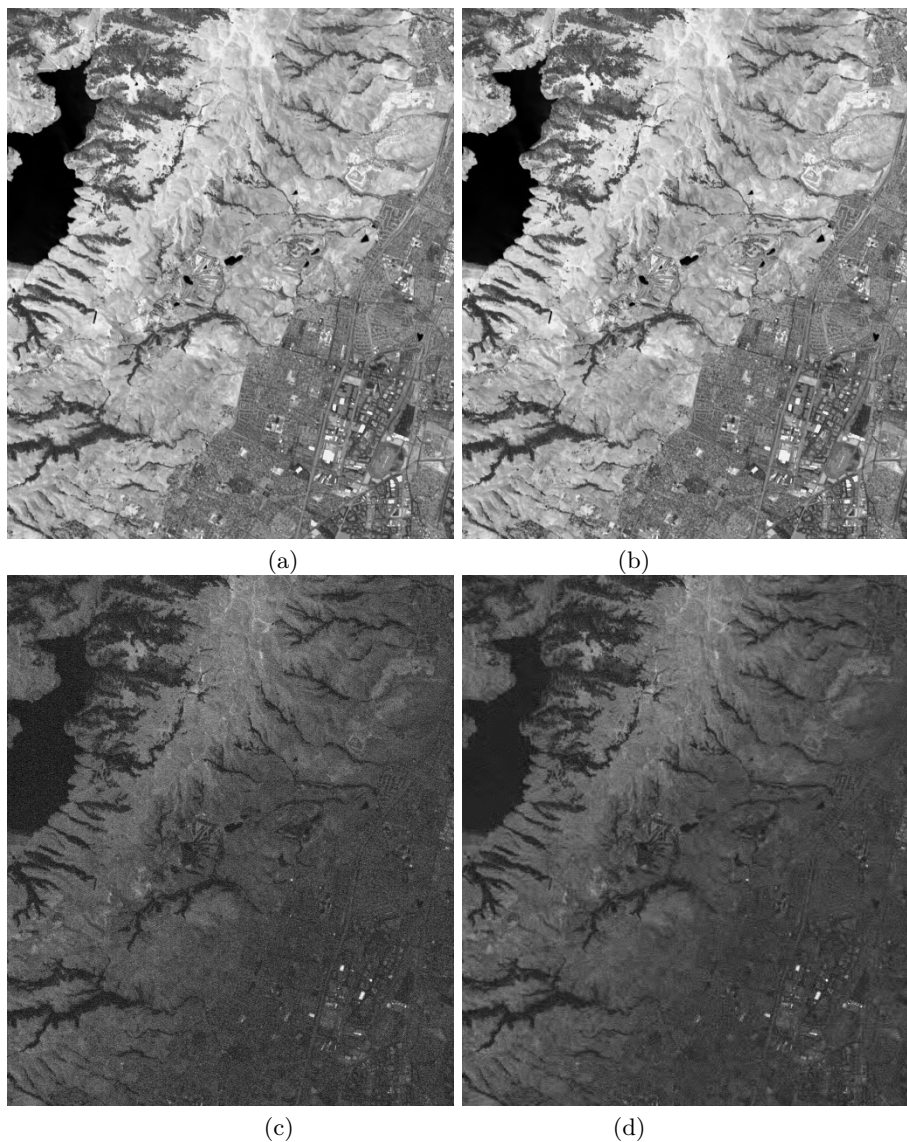


Fig. 4. The noisy sub-band image (Moffett Field, channel 128): the original (a) one and the image after compression/decompression (b). The sub-band image with lower SNR (Moffett Field, channel 223): the original (c) one and the image after compression/decompression (d).

$QS = 4.5$. The values $\hat{\sigma}_n^2$, $n = 1, \dots, 224$ or, better, $\hat{\sigma}_n$, $n = 1, \dots, 224$ are coded in a lossless manner and passed as side information to be used at the final stage of the decompression.

The decompression has to be performed in the inverse order. After the first stage, the normalized decompressed images are obtained. Then, they are scaled

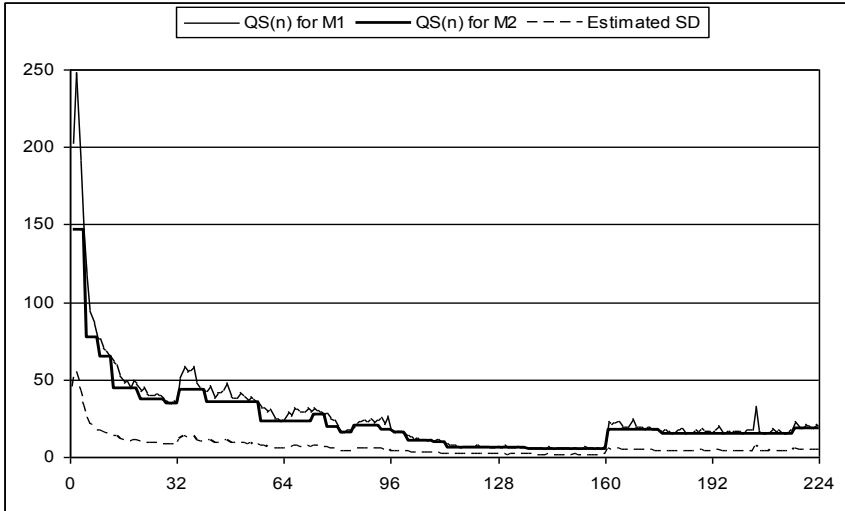


Fig. 5. Estimates of $\hat{\sigma}_n$ for the Jasper Ridge images and $QS(n)$ for compression methods M1 and M2

Table 1. The compression ratio (CR) for the method M2 [16] and the proposed method (M3)

Hyperspectral Image	CR for M2	CR for M3	
		16 sub-bands	32 sub-bands
Cuprite	20.68	31.30	34.17
Jasper Ridge	9.94	10.95	9.08
Lunar Lake	24.39	33.50	34.87
Moffett Field	8.95	9.58	8.01

by multiplying each sub-band image values by $\hat{\sigma}_n$ taken from the side information for the decompression.

Since the 3D AGU coder is based on the DCT, it is worth using the group size proportional to the power of 2 to provide fast coding. These can be, e.g., group sizes Q equal to 8, 16, or 32 (in the latest case, 224 sub-bands of the AVIRIS data are divided into 7 groups).

The proposed method (M3) has been tested for four hyperspectral AVIRIS images ($QS = 4.5$ for all groups, $\sigma^2 \approx 1$). We analyzed two cases: 16 and 32 sub-band images in each group. The obtained compression ratios are presented in Table 1 above. It is seen that for 16 sub-bands in the group the compression ratio increased for all four test images. The largest increase is observed for less complex images, Cuprite and Lunar Lake (CR has increased by almost 50 %). For images with more complex structure (Jasper Ridge and Moffett Field), CR has increased by about 6 . . . 10%. If the number of sub-bands in the group is 32, CR has increased (in comparison to the method M2) for images with a simpler

structure but reduced for the images Jasper Ridge and Moffett Field. Thus, our recommendation is to use 16 sub-bands in the group.

4 Conclusions

The modification of the DCT-based method for the lossy compression of the hyperspectral AVIRIS data is proposed. Due to the proposed blind estimation of the additive noise standard deviation and the image normalization with sub-band grouping, the compression ratio (CR) has increased by 6 ... 50% with respect to the method designed earlier. The grouping of the bands remains image-dependent: images with simple structures allow a higher number of bands in a group. If there are more complex structures then the compression ratio may reduce. Thus, the recommendation is to use a quasi-optimal number, namely 16, of the bands in a group to remain in the safe side with respect to the compression. The filtering effect is observed for sub-bands with low SNR whilst no visual distortions take place for sub-bands with large SNR.

References

1. Christophe, E., Leger, D., Mailhes, C.: Quality criteria benchmark for hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing* 43(9), 2103–2114 (2005)
2. Chang, C.-I. (ed.): *Hyperspectral Data Exploitation: Theory and Applications*. Wiley-Interscience, Hoboken (2007)
3. Kaarna, A.: Compression of Spectral Images. In: Obinata, G., Dutta, A. (eds.) *Vision Systems: Segmentation and Pattern Recognition*, I-Tech, Austria, ch. 14, pp. 269–298 (2007)
4. Mielikainen, J.: Lossless compression of hyperspectral images using lookup tables. *IEEE Signal Processing Letters* 13(3), 157–160 (2006)
5. Penna, B., Tillo, T., Magli, E., Olmo, G.: Transform coding techniques for lossy hyperspectral data compression. *IEEE Transactions on Geoscience and Remote Sensing* 45(5), 1408–1421 (2007)
6. Aiazzi, B., Baronti, S., Lastri, C., Santurri, L., Alparone, L.: Low complexity lossless/near-lossless compression of hyperspectral imagery through classified linear spectral prediction. In: *Proceedings of IEEE International Geoscience and Remote Sensing Symposium*, p. 4 (2005)
7. Tang, X., Pearlman, W.A.: Three-dimensional wavelet-based compression of hyperspectral images. In: Motta, G., Rizzo, F., Storer, J.A. (eds.) *Hyperspectral Data Compression*, pp. 273–308. Springer US, Heidelberg (2006)
8. Miguel, A.C., Askew, A.R., Chang, A., Hauck, S., Ladner, R.E., Riskin, E.A.: Reduced complexity wavelet-based predictive coding of hyperspectral images for FPGA implementation. In: *Proceedings of Data Compression Conference*, pp. 1–10 (2004)
9. Ryan, M.J., Pickering, M.R.: An improved M-NVQ algorithm for the compression of hyperspectral data. In: *Proceedings of IEEE International Geoscience and Remote Sensing Symposium*, vol. 2, pp. 600–602 (2000)

10. Du, Q., Fowler, J.E.: Hyperspectral Image Compression Using JPEG2000 and Principal Component Analysis. *IEEE Transactions on Geoscience and Remote Sensing* 4(2), 201–205 (2007)
11. Cagnazzo, M., Poggi, G., Verdoliva, L., Zinicola, A.: Region-oriented compression of multispectral images by shape-adaptive wavelet transform and SPIHT. In: *Proceedings of IEEE International Conference on Image Processing*, pp. 2459–2462 (2004)
12. Penna, B., Tillo, T., Magli, E., Olmo, G.: Progressive 3-D Coding of Hyperspectral Images Based on JPEG2000. *IEEE Geoscience and Remote Sensing Letters* 3(1), 125–129 (2006)
13. Du, Q., Chang, C.-I.: Linear mixture analysis-based compression for hyperspectral image analysis. *IEEE Transactions on Geoscience and Remote Sensing* 42(4), 875–891 (2004)
14. Bovik, A.: *Handbook on Image and Video Processing*. Academic Press, USA (2000)
15. Curran, P.J., Dungan, J.L.: Estimation of signal-to-noise: a new procedure applied to AVIRIS data. *IEEE Transactions on Geoscience and Remote Sensing* 27(7), 620–628 (1989)
16. Ponomarenko, N., Lukin, V., Zriakhov, M., Kaarna, A., Astola, J.: An automatic approach to lossy compression of AVIRIS images. In: *Proceedings of IEEE International Geoscience and Remote Sensing Symposium IGARSS 2007*, pp. 472–475 (2007)
17. White, R.L., Percival, J.W.: Compression and Progressive Transmission of Astronomical Images. In: *SPIE Proceedings*, vol. 2199, pp. 703–713 (1994)
18. Du, Q., Fowler, J.E., Zhu, W.: On the Impact of Atmospheric Correction on Lossy Compression of Multispectral and Hyperspectral Imagery. *IEEE Transactions on Geoscience and Remote Sensing* 47(1), 130–132 (2009)
19. Ponomarenko, N., Lukin, V., Zriakhov, M., Egiazarian, K., Astola, J.: Estimation of accessible quality in noisy image compression. In: *Proceedings of EUSIPCO*, p. 4 (2006)
20. Yea, S., Pearlman, W.A.: Critical encoding rate in combined denoising and compression. In: *Proceedings of IEEE International Conference on Image Processing*, vol. III, p. 4 (2005)
21. Fidler, A., Skaleric, U.: The impact of image information on compressibility and degradation of medical image compression. *Med. Phys.* 33(8), 2832–2839 (2006)
22. Lukin, V., Vozel, B., Abramov, S., Ponomarenko, N., Uss, M., Chehdi, K.: Blind methods for noise evaluation in multi-component images. In: Collet, C., Chanussot, J., Chehdi, K. (eds.) *Multivariate image processing*, ISTE Ltd., France, ch. 9, pp. 263–302 (2009)
23. Ponomarenko, N., Lukin, V., Egiazarian, K., Astola, J.: A method for blind estimation of spatially correlated noise characteristics. In: *Proceedings of SPIE Image Processing: Algorithms and Systems VIII SPIE*, vol. 7532, p. 12 (2010)
24. Al-Chaykh, O.K., Mersereau, R.M.: Lossy compression of noisy images. *IEEE Transactions on Image Processing* 7(12), 1641–1652 (1998)
25. Lukin, V.V., Ponomarenko, N.N., Zelensky, A.A., Kurekin, A.A., Lever, K.: Compression and classification of noisy multichannel remote sensing images. In: *Proceedings of Image and Signal Processing for Remote Sensing XIV*, Cardiff, UK, SPIE, vol. 7109, p. 12 (2008)
26. Ponomarenko, N.N., Lukin, V.V., Egiazarian, K., Astola, J.: DCT based high quality image compression. In: *Proceedings of 14th Scandinavian Conference on Image Analysis*, pp. 1177–1185 (2005)

27. Lukin, V., Abramov, S., Uss, M., Marusiy, I., Ponomarenko, N., Zelensky, A., Vozel, B., Chehdi, K.: Testing of methods for blind estimation of noise variance on large image database. In: Marchuk, V.I. (ed.) *Theoretical and Practical Aspects of Digital Signal Processing in Information-Communication Systems*, Shakhty, Russia, ch. 2, pp. 43–70 (2009), <http://k504.xai.edu.ua/html/prepods/lukin/BookCh1.pdf>
28. Ponomarenko, N., Lukin, V., Zriakhov, M., Kaarna, A.: Two aspects in lossy compression of hyperspectral AVIRIS images. In: *Proceedings of MMET*, Odessa, Ukraine, pp. 375–377 (June 2008)

New Saliency Point Detection and Evaluation Methods for Finding Structural Differences in Remote Sensing Images of Long Time-Span Samples

Andrea Kovacs¹ and Tamas Sziranyi²

¹ Pazmany Peter Catholic University
Prater 50/A, 1083, Budapest, Hungary

² Hungarian Academy of Sciences, Computer and Automation Research Institute
Distributed Events Analysis Research Group
Kende 13-17, 1111, Budapest, Hungary
{kovacs.andrea,sziranyi}@sztaki.hu

Abstract. The paper introduces a novel methodology to find changes in remote sensing image series. Some remotely sensed areas are scanned frequently to spot relevant changes, and several repositories contain multi-temporal image samples for the same area. The proposed method finds changes in images scanned by a long time-interval difference in very different lighting and surface conditions. The presented method is basically an exploitation of Harris saliency function and its derivatives for finding featuring points among image samples. To fit together the definition of keypoints and their active contour around them, we have introduced the Harris corner detection as an outline detector instead of the simple edge functions. We also demonstrate a new local descriptor by generating local active contours. Saliency points support the boundary hull definition of objects, constructing by graph based connectivity detection and neighborhood description. This graph based shape descriptor works on the saliency points of the difference and in-layer features. We prove the method in finding structural changes on remote sensing images.

Keywords: remote sensing, Harris function, change detection.

1 Introduction

Automatic evaluation of aerial photograph repositories is an important field of research since manual administration is time consuming and cumbersome. Long time-span surveillance or reconnaissance about the same area can be crucial for quick and up-to-date content retrieval. The extraction of changes may facilitate applications like urban development analysis, disaster protection, agricultural monitoring, and detection of illegal garbage heaps, or wood cuttings. The obtained change map should provide useful information about size, shape, or quantity of the changed areas, which could be applied directly by higher level object

analyzer modules [1], [2]. While numerous state-of-the-art approaches in remote sensing deal with multispectral [3], [4], [5], [6] or synthetic aperture radar (SAR) [7], [8] imagery, the significance of handling optical photographs is also increasing [9]. Here, the processing methods should consider that several optical image collections include partially archive data, where the photographs are grayscale or contain only poor color information. This paper focuses on finding contours of newly appearing/fading out objects in optical aerial images which were taken with several years time differences partially in different seasons and in different lighting conditions. In this case, simple techniques like thresholding the difference image [10] or background modeling [11] cannot be adopted efficiently since details are not comparable.

These optical image sensors provide limited information and we can only assume to have image repositories which contain geometrically corrected and registered [12] grayscale orthophotographs.

In the literature one main group of approaches is the postclassification comparison, which segments the input images with different land-cover classes, like arboreous lands, barren lands, and artificial structures [13], obtaining the changes indirectly as regions with different classes in the two image layers [9]. We follow another methodology, like direct methods [3], [5], [5], [7], where a similarity-feature map from the input photographs (e.g., a DI) is derived, then the feature map is separated into changed and unchanged areas.

Our direct method does not use any land-cover class models, and attempts to detect changes which can be discriminated by low-level features. However, our approach is not a pixel-neighborhood MAP system as in [14], but a connection system of nearby saliency points. These saliency points define a connection system by using local graphs for outlining the local hull of the objects. Considering this curve as a starting spline, we search for objects' boundaries by active contour iterations.

The above features are local saliency points and saliency functions. The main saliency detector is calculated as a discriminative function among the functions of the different layers. We show that Harris detector is the appropriate function for finding the dissimilarities among different layers, when comparison is not possible because of the different lighting, color and contrast conditions.

Local structure around keypoints is investigated by generating scale and position invariant descriptors, like SIFT. These descriptors describe the local microstructure, however, in several cases more succinct set of parameters is needed. For this reason we have developed a local active-contour based descriptor around keypoints, but this contour is generated by edginess in the cost function, while we characterize keypoints of junctions. To fit together the definition of keypoints and their active contour around them, we have introduced the Harris corner detection as an outline detector instead of the simple edge functions. This change resulted in a much better characterization of local structure.

In the following, we introduce a new change detection procedure by using Harris function and its derivatives for finding saliency points among image samples; then a new local descriptor will be demonstrated by generating local active

contours; A graph based shape descriptor will be shown based on the saliency points of the difference and in-layer features; finally, we prove the methods capabilities for finding structural changes on remote sensing images.

2 Change Detection with Harris Keypoints

2.1 Harris Corner Detector

The detector was introduced by Chris Harris and Mike Stephens in 1988 [15]. The algorithm based on the principle that at corner points intensity values change largely in multiple directions. By considering a local window in the image and determining the average changes of image intensity result from shifting the window by a small amount in various directions, all the shifts will result in large change in case of a corner point. Thus corner can be detected by finding when the minimum change produced by any of shifts is large.

The method first computes the Harris matrix (M) for each pixel in the image. Then, instead of computing the eigenvalues of M , an R corner response is defined:

$$R = Det(M) - k * Tr^2(M) \quad (1)$$

This R characteristic function is used to detect corners. R is large and positive in corner regions, and negative in edge regions. By searching for local maximas of a normalized R , the Harris keypoints can be found. Normalizing makes R smoother and only major corner points are detected. R could also be used for edge detection: $|R|$ function is large and positive in corner and also positive but smaller in edge regions, and nearly zero in flat regions. We used this function in our later work. Figure 1 shows the result of Harris keypoint detection. On Figure 1(b) light regions shows the larger R values, so keypoints will be detected in these areas (Figure 1(c)).

2.2 Change Detection

The advantage of Harris detector is its strong invariance to rotation and the R characteristic function's invariance to illumination variation and image noise. Therefore it could be used efficiently for change detection in airborne images. In

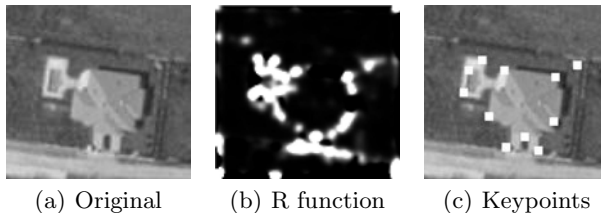


Fig. 1. Operation of Harris detector: Corner points are chosen as the local maximas of the R characteristic function

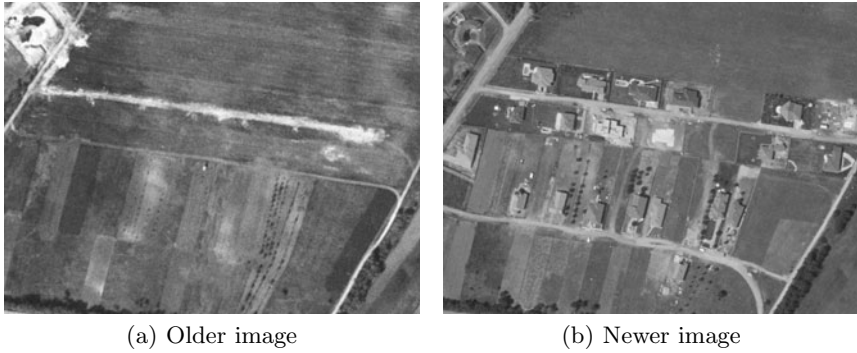


Fig. 2. Original image pairs

these kind of images, changes can mean the appearance of new man-made objects, (like buildings or streets), or natural, environmental variations. As image pairs may be taken with large intervals of time, the area may change largely. In our case the pieces of the image pairs was taken in 2000 and 2005 (Figure 2). The registration was performed manually by Hungarian Institute of Geodesy, Cartography and Remote Sensing, therefore we worked on image pairs representing exactly the same area.

In our work we mainly focus on newly built objects (buildings, pools, etc.). There are many difficulties when detecting such objects in airborne images. The illumination and weather circumstances may vary, resulting different colour, contrast and shadow conditions. The urban area might be imaged from different point of view. Buildings can be hidden by other structures, like trees, shadows or other buildings. These objects are quite various, which also makes the detection tough.

To overcome a part of the aforementioned difficulties, our idea was to use the difference of the image pairs. As we are searching for newly built objects, we need to find buildups, that only exist on the newer image, therefore having large effect both in the difference image and the newer image.

First, we examined the usability of intensity based (Figure 3(a)) and edge based difference map (Figure 3(b)).

Intensity based and edge based difference maps are calculated as follows:

$$I_{\text{diff}}^{\text{mod}} = |I_{\text{old}}^{\text{mod}} - I_{\text{new}}^{\text{mod}}| \quad (2)$$

where I_{old} and I_{new} means the older and newer pieces of the image pairs respectively. The upper index mod refers to the basis of the modification: for example in case of the edginess $I_{\text{new}}^{\text{mod}} = I_{\text{new}}^{\text{edge}} = \text{edge}(I_{\text{new}})$.

When searching for keypoint candidates, we call for Harris detector. As written before, the new objects have high effects both on the new and difference, therefore we search for such keypoints that accomplish the next two criterias simultaneously:

1. $R(I_{\text{diff}}^{\text{mod}}) > \epsilon_1$
2. $R(I_{\text{new}}^{\text{mod}}) > \epsilon_2$

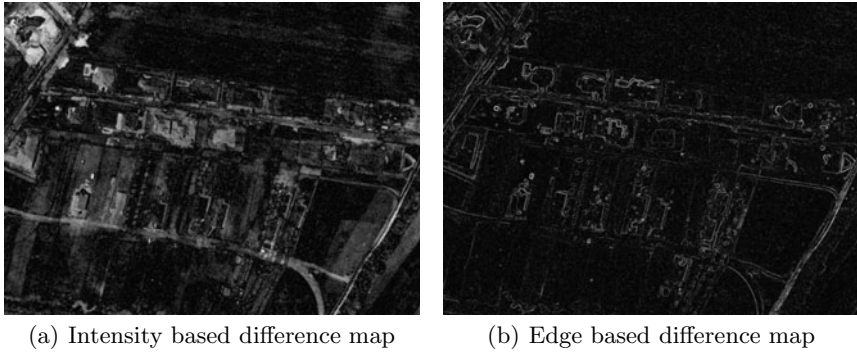


Fig. 3. Difference maps

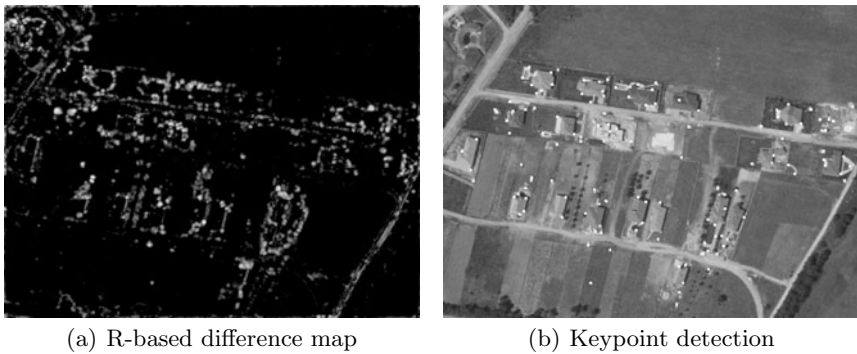


Fig. 4. Logarithmized difference map and result of detection based on the R-function

$R(\dots)$ indicates the Harris characteristic function (Eq. 1), ϵ_1 and ϵ_2 are thresholds. It is advised to take smaller ϵ_2 , than ϵ_1 . With this choice the difference map is preferred and has larger weight. Only important corners in the difference map will be marked.

After performing keypoint candidate detection on intensity based and edge based difference maps, we determined that both of them are too sensitive to illumination change, so altering contrast and color conditions result the appearance of false edges and corner points and the vanishing of real ones in the difference map.

Therefore, we decided to use another metric instead of intensity and edginess and redefine the difference map according to the new metric. The chosen metric was the Harris R characteristic function. Therefore the difference map was calculated as:

$$I_{\text{diff}}^R = |R_{\text{old}} - R_{\text{new}}| \tag{3}$$

Modification of I_{new} looks as $I_{\text{new}}^R = R_{\text{new}}$.

The logarithm of difference map is in Figure 4(a). As R-function has lower values, the image can be better seen, if the natural logarithm is illustrated instead of the original map.

The keypoint candidate detection method was the same as written before. Results are in Figure 4(b). Keypoint candidates cover all buildings, and only a few points are in false areas. The false candidates have to be filtered out with further techniques, described in the next section.

3 Object Contour Detection with Saliency Functions and Graph Theory

3.1 Detection of Local Structures

According to [16], local contours around keypoints are efficient, low dimensional tool for matching and distinguishing, therefore this algorithm was now implemented for Harris keypoint candidates to filter out the falsely detected points. The main steps for estimating local structure characteristics:

1. Generating Harris keypoints for difference map (Section 2.2)
2. Generating the Local Contour around keypoints in the original image [17]
3. Calculating Modified Fourier Descriptor for the estimated closed curve [18]
4. Describe the contour by a limited set of Fourier coefficients [19]

As the specification shows, after detecting the Harris keypoint candidates (the method is briefly summarized in Section 2.2), GVF Snake [17] was used for local contour (LC) analysis. LC was computed in the original image, in a 20×20 size area, where the keypoint was in the middle. The generated LC assigns an individual shape to every keypoint, but the dimension is quite high. Therefore modified Fourier descriptors were applied, which represents the LC at low dimension. We have determined the cut-off frequency by maximizing the recognition accuracies and minimizing the noise of irregularities of the shape boundaries and chose the first twenty coefficients (excluding the DC component to remove the positional sensitivity).

3.2 Matching with Local Contours

Our assumption was that after having the FDs for the keypoints, differences between keypoint surroundings can be searched through this descriptor set. We extended the MFD method to get symmetric distance computation as it is written in [16]. By comparing a keypoint (p_i) on the first frame and on the second frame, D_i represents the similarity value. If the following criteria exists:

$$D_i > \epsilon_3 \quad (4)$$

where $\epsilon_3 = 3$ is a tolerance value, than the keypoint is supposed to be a changed area.

Results of the detection is provided in Figure 5(a).

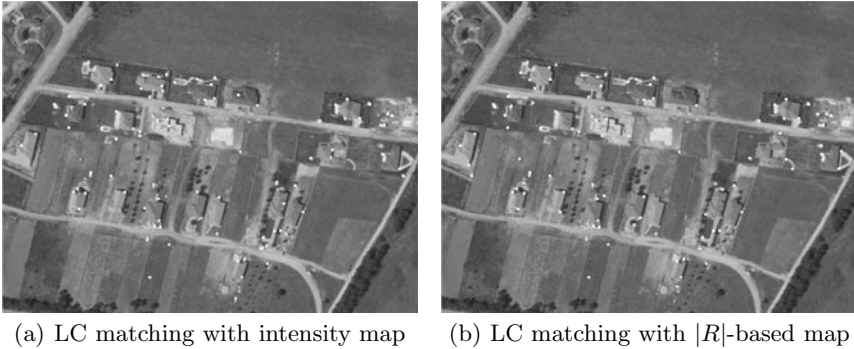


Fig. 5. Result of Local Contour matching with different edge maps

3.3 New Edge Map

As it was also written in Section 2.2 intensity based edge maps are sensitive to illumination changes. This problem also occurred when comparing local features: changeless places were declared as newly built objects. Therefore we used the Harris based feature map, denoted by R_{old} and R_{new} in Section 2.2:

$$f_{|R|}(x, y) = G_{\sigma}(x, y) * |R(x, y)| \quad (5)$$

Detected contours are smoother and more robust in case of the $|R|$ function. We benefit from this smoothness, as contours can be distinguished easier. However, as there might be no real contour in the neighbourhood of the keypoints, AC-method is only used for exploiting the local information to get low-dimensional descriptor, therefore significance of accuracy is overshadowed by efficiency of comparison. The detected points based on the $|R|$ -function can be seen in Figure 5(b).

3.4 Enhancing the Number of Saliency Points

After selecting the saliency points (or keypoints) indicating change, we now have to enhance the number of keypoints. Therefore we are looking for saliency points that are not presented in the older image, but exists on the newer one. We call for the Harris corner detection method again. By calculating Harris corner points for older and newer image as well, an arbitrary $q_i = (x_i, y_i)$ point is selected if it satisfies all of the following conditions:

- (1.) $q_i \in H_{\text{new}}$
- (2.) $q_i \notin H_{\text{old}}$
- (3.) $d(q_i, p_j) < \epsilon_4$

H_{new} and H_{old} are the sets of Harris keypoints generated in the newer and older image, $d(q_i, p_j)$ is the Euclidean distance of q_i and p_j , where p_j denotes the point with smallest Euclidean-distance to q_i selected from H_{old} .

New points are searched iteratively, with $\epsilon_4 = 10$ condition. Here, ϵ_4 depends on the resolution of the image and on the size of buildings. If resolution is smaller, than ϵ_4 has to be chosen as a smaller value.

Figure 6 shows the enhanced number of keypoints.



Fig. 6. Enhanced number of Harris keypoints

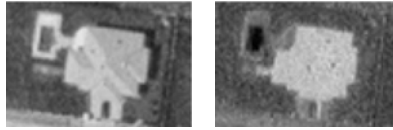


Fig. 7. Grayscale images generated two different ways: (a) is the R component of the RGB color space, (b) is the u^* component of the $L^*u^*v^*$ color space

3.5 Reconciling Edge Detection and Corner Detection

Now an enhanced set of saliency points is given, denoting possible area of changes, which serves as the basis for building detection. We redefine the problem in terms of graph theory [20].

A graph G is represented as $G = (V, E)$, where V is the vertex set, E is the edge network. In our case, V is already defined by the enhanced set of Harris points. Therefore, E needs to be formed.

Information about how to link the vertices can be gained from edge maps. These maps can help us to only match vertices belonging to the same building.

If objects have sharp edges, we need such image modulations, which emphasize these edges as strong as it is possible. By referring to Figure 7(a) and 7(b), we can see that R component of RGB and u^* component of $L^*u^*v^*$ color space can intensify building contours. Both of them operates suitably in different cases, therefore we apply both.

By generating the R and u^* components (further on denoted as $I_{new,r}$ and $I_{new,u}$) of the original, newer image, Canny edge detection [21] with large threshold ($Thr = 0.4$) is executed on them. $C_{new,r}$ and $C_{new,u}$ marks the result of Canny detection. (Figure 8(a) and 8(b))

The process of matching is as follows. Given two vertices: $v_i = (x_i, y_i)$ and $v_j = (x_j, y_j)$. We match them if they satisfy the following conditions:

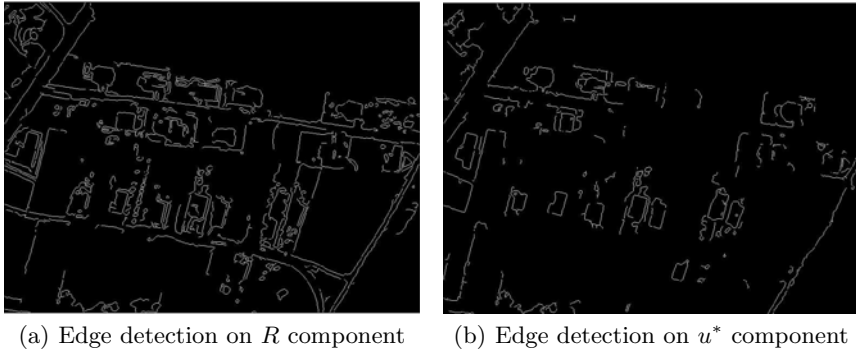


Fig. 8. Result of Canny edge detection on different colour components



Fig. 9. Subgraphs given after matching procedure

- (1.) $d(v_i, v_j) = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} < \epsilon_5$,
- (2.) $C_{\text{new}, \dots}(x_i, y_i) = \text{true}$,
- (3.) $C_{\text{new}, \dots}(x_j, y_j) = \text{true}$,
- (4.) \exists a finite path between v_i and v_j .

$C_{\text{new}, \dots}$ indicates either $C_{\text{new}, r}$ or $C_{\text{new}, u}$. ϵ_5 is a tolerance value, which depends on the resolution and average size of the objects. We apply $\epsilon_5 = 30$.

These conditions guarantee that only vertices connected just in the newer edge map are matched. Like in the lower right part of Figure 9 two closely located buildings are separated correctly.

We obtain a graph composed of many separate subgraphs, which can be seen in Figure 9. Each of these connected subgraph is supposed to represent a building. However, there might be some unmatched keypoints, indicating noise. To discard them, we select subgraphs having at least two vertices.

To determine the contour of the subgraph-represented buildings, we used the aforementioned GVF snake method. The convex hull of the vertices in the subgraphs is applied as the initial contour.

4 Experiments and Conclusion

Some results of the contour detection can be seen in Figure 10.

The algorithm was tested for a few registered image pairs and the results were promising. The algorithm was able to find almost every changed building and to filter out non-changed ones.

The main advantage of our method is that it does not need any building template and can detect objects of any size and shape. The method has difficulties in finding objects with similar colour to the background and sometimes one object is described with more than one subgraphs. These problems need to be solved in a forthcoming semantic or object evaluation step.

Harris characteristic function was used to determine changes between registered image pairs scanned with long time interval. The detected keypoint candidates were then filtered and the number of remaining keypoints was enhanced by saliency methods. A graph based representation was used to create initial contour of changed objects, then GVF snake method generated the object boundaries. Our experiments showed that saliency methods can be efficient tools when determining changes. Our future works includes more evaluation and comparison with other state-of-the-art algorithms.

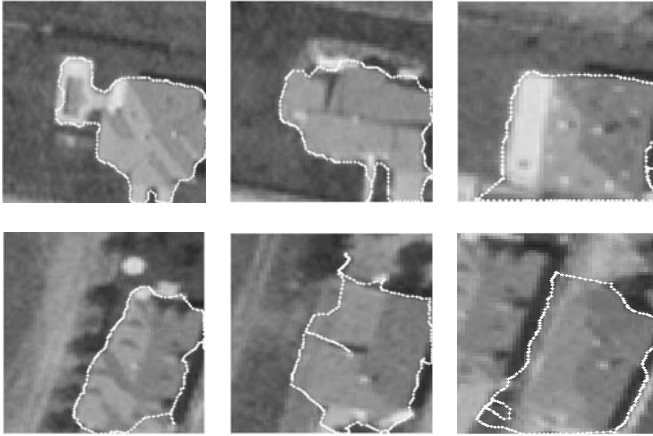


Fig. 10. Results of the contour detection by GVS snake method. The initial contour was the convex hull of the vertices in the subgraph representing the object.

Acknowledgements

This work was supported by the Hungarian Scientific Research Fund under grant number 76159. The authors would like to thank for the registered image pairs to Hungarian Institute of Geodesy, Cartography and Remote Sensing (FÖMI).

References

1. Peng, T., Jermyn, I.H., Prinet, V., Zerubia, J.: Incorporating generic and specific prior knowledge in a multi-scale phase field model for road extraction from vhr images. *IEEE Trans. Geoscience and Remote Sensing* 1(2), 139–146 (2008)
2. Lafarge, F., Descombes, X., Zerubia, J.B., Deseilligny, M.P.: Automatic building extraction from dems using an object approach and application to the 3d-city modeling. *ISPRS Journal of Photogrammetry and Remote Sensing* 63(3), 365–381 (2008)
3. Ghosh, S., Bruzzone, L., Patra, S., Bovolo, F., Ghosh, A.: A context-sensitive technique for unsupervised change detection based on hopfield-type neural networks. *IEEE Trans. Geoscience and Remote Sensing* 45(3), 778–789 (2007)
4. Perrin, G., Descombes, X., Zerubia, J.: 2d and 3d vegetation resource parameters assessment using marked point processes. In: *Proc. of the 18th International Conference on Pattern Recognition*, pp. 1–4 (2006)
5. Wiemker, R.: An iterative spectral-spatial bayesian labeling approach for unsupervised robust change detection on remotely sensed multispectral imagery. In: *Proc. of Conference on Computer Analysis of Images and Patterns*, pp. 263–270 (1997)
6. Bruzzone, L., Prieto, D.F.: An adaptive semiparametric and context-based approach to unsupervised change detection in multitemporal remote-sensing images. *IEEE Trans. Image Processing* 11(4), 452–466 (2002)
7. Bazi, Y., Bruzzone, L., Melgani, F.: An unsupervised approach based on the generalized gaussian model to automatic change detection in multitemporal sar images. *IEEE Trans. Geoscience and Remote Sensing* 43(4), 874–887 (2005)
8. Gamba, P., Dell’Acqua, F., Lisini, G.: Change detection of multitemporal sar data in urban areas combining feature-based and pixel-based techniques. *IEEE Trans. Geoscience and Remote Sensing* 44(10), 2820–2827 (2006)
9. Zhong, P., Wang, R.: A multiple conditional random fields ensemble model for urban area detection in remote sensing optical images. *IEEE Trans. Geoscience and Remote Sensing* 45(12), 3978–3988 (2007)
10. Benedek, C., Szirányi, T., Kato, Z., Zerubia, J.: Detection of object motion regions in aerial image pairs with a multilayer markovian model. *IEEE Trans. Image Processing* 18(10), 2303–2315 (2009)
11. Benedek, C., Szirányi, T.: Bayesian foreground and shadow detection in uncertain frame rate surveillance videos. *IEEE Trans. Image Processing* 17(4), 608–621 (2008)
12. Shah, C.A., Sheng, Y., Smith, L.C.: Automated image registration based on pseudo-invariant metrics of dynamic land-surface features. *IEEE Trans. Geoscience and Remote Sensing* 46(11), 3908–3916 (2008)
13. Castellana, L., d’Addabbo, A., Pasquariello, G.: A composed supervised/unsupervised approach to improve change detection from remote sensing. *IEEE Pattern Recognition Letters* 28(4), 405–413 (2007)
14. Benedek, C., Szirányi, T.: Change detection in optical aerial images by a multilayer conditional mixed markov model. *IEEE Trans. Geoscience and Remote Sensing* 47(10), 3416–3430 (2009)
15. Harris, C., Stephens, M.: A combined corner and edge detector. In: *Proc. of the 4th Alvey Vision Conference*, pp. 147–151 (1988)
16. Kovacs, A., Sziranyi, T.: Local contour descriptors around scale-invariant keypoints. In: *Proc. of IEEE International Conference on Image Processing, Cairo, Egypt*, pp. 1105–1108 (2009)

17. Xu, C., Prince, J.L.: Gradient vector flow: A new external force for snakes. In: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp. 66–71 (1997)
18. Rui, Y., She, A., Huang, T.S.: A modified fourier descriptor for shape matching in MARS. In: Image Databases and Multimedia Search, pp. 165–180 (1998)
19. Licsar, A., Sziranyi, T.: User-adaptive hand gesture recognition system with interactive training. *Image and Vision Computing* 23(12), 1102–1114 (2005)
20. Sirmacek, B., Unsalan, C.: Urban-area and building detection using sift keypoints and graph theory. *IEEE Trans. Geoscience and Remote Sensing* 47(4), 1156–1167 (2009)
21. Canny, J.: A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence* 8(6), 679–698 (1986)

Regularized Kernel Locality Preserving Discriminant Analysis for Face Recognition

Xiaohua Gu, Weiguo Gong, Liping Yang, and Weihong Li

Laboratory of Optoelectronic Technology and Systems of the Education Ministry of China, Chongqing University, Chongqing, 400044 China
{xhgu, wggong, yanglp, weihongli}@ccqu.edu.cn

Abstract. In this paper, a regularized kernel locality preserving discriminant analysis (RKLPDA) method is proposed for facial feature extraction and recognition. The proposed RKLPDA comes into the characteristic of LPDA that encodes both the geometrical and discriminant structure of the data manifold, and improves the classification ability for linear non-separable data by introducing kernel trick. Meanwhile, by regularizing the eigenvectors of the kernel locality preserving within-class scatter, RKLPDA utilizes all the discriminant information and eliminates the small sample size (SSS) problem. Experiments on ORL and FERET face databases are performed to test and evaluate the proposed algorithm. The results demonstrate the effectiveness of RKLPDA.

Keywords: Locality preserving discriminant analysis, Kernel method, Feature extraction, Face recognition.

1 Introduction

Discriminant analysis is a technique of finding a transformation which characterizes or separates two or more classes by maximizing the inter-class diversity and meanwhile minimizing the intra-class compactness. Representative discriminant analysis methods include linear discriminant analysis (LDA) [1], locality preserving discriminant analysis (LPDA) [2], and their null space extensions, null space LDA (NLDA) [3], null space DLPP (NDLPP) [4] and etc.. LDA based methods, which dwell on estimating the global statistics, fail to discover the underlying structure if the data lie on or close to a sub-manifold embedding in the high-dimensional input space. LPDA based methods, as the discriminant analysis extensions of locality preserving projections (LPP) [5], encode both the geometrical and discriminant structure of the data manifold and are more powerful. However, when applied to face recognition, they may suffer from the following problems: (1) Due to the high dimensionality of the sample space and the limited training samples, LDA and LPDA always suffer from the well-known SSS problem; (2) The discriminative information resides in both the principal and the null subspaces of intra-class compactness matrix [6]. Nevertheless, NLDA and NLPDA extract only that in the null subspace; (3) For C -class recognition

task, the number of features obtained by all the aforementioned methods has an upper limit $C - 1$, which is often insufficient to separate the classes well.

Kernel methods [7] [8] [9], which provide powerful extensions of linear methods to nonlinear cases by performing linear operations on higher or even infinite dimensional features transformed by a kernel mapping function, have been widely researched. The kernel-based formulations of many linear subspace methods have been proposed so far, including kernel PCA (KPCA) [10], kernel fisher discriminant analysis (KFDA) [7], kernel class-wise LPP (KCLPP) [8] and etc.. Most of these kernel-based methods outperform their corresponding linear cases for face recognition. However, the kernel extensions of the discriminant analysis methods, called kernel discriminant analysis methods, also suffer from the aforementioned problems.

In this paper, we derive the kernel locality preserving discriminant analysis (KLPDA) by introducing kernel trick to improve the classification ability of LPDA on linear non-separable face images. To address the above problems, a regularization procedure is then employed, by which the eigenvectors of the kernel locality preserving within-class scatter matrix are weighted according to the corresponding predicted eigenvalues, and finally discriminant features are extracted in the regularized subspace spanned by the weighted eigenvectors. In predicting eigenvalues, the small ones which are suspicious to sample noises are raised and the zeros are set to a small constant. Through this procedure, the entire space including the principal subspace and the null subspace is utilized to extract the discriminant features, even the null space is highlighted. And also, the regularized kernel locality preserving within-class scatter matrix is nonsingular, hence the SSS problem is eliminated and the number of final features obtained by RKLPDA is extended to $n - 1$, where n is number of training samples.

2 Kernel LPDA (KLPDA)

2.1 Schema of LPDA

LPDA tries to find a linear transformation to project the high-dimensional dataset to a low-dimensional embedding, which preserves the local neighborhood relationship of samples and meanwhile enhances the separability of samples. Given a set of N -dimensional face image samples $X = \{x_i\}_{i=1}^n$, $x_i \in \mathbb{R}^{N \times 1}$ from C classes, the linear transformation $A \in \mathbb{R}^{d \times n}$, where d is the reduced dimensionality, can be obtained by maximizing an objective function as follows:

$$J(A) = \frac{\sum_{i,j=1}^C (m_i - m_j) b_{ij} (m_i - m_j)^T}{\sum_{k=1}^C \sum_{y_i, y_j \in \omega_k, y_i \neq y_j} (y_i - y_j) w_{ij}^k (y_i - y_j)^T} \tag{1}$$

where $y_i = A^T x_i$, $m_i = (1/n_i) \sum_{y_k \in \omega_i} y_k$ are the low-dimensional embedding and class mean vector, n_i is the number of samples in class ω_i , $\sum_{i=1}^C n_i = n$. The weight matrices $W = \{diag([w_{ij}^k]_{i,j=1}^{n_k})\}_{k=1}^C$ and $B = [b_{ij}]_{i,j=1}^C$ are constructed

by the neighborhood relationships and labels of samples. Their components are defined as

$$w_{ij}^k = \begin{cases} \exp(-\frac{\|x_i - x_j\|^2}{2t^2}), & x_i \in NN_p(x_j) \text{ or } x_j \in NN_p(x_i), x_i, x_j \in \omega_k. \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

$$b_{ij} = \begin{cases} \exp(-\frac{\|u_i - u_j\|^2}{2t^2}), & u_i \in NN_p(u_j) \text{ or } u_j \in NN_p(u_i). \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

where parameter t is a suitable constant, $u_i = (1/n_i) \sum_{x_k \in \omega_i} x_k$ is the mean vector of class i in input space, and $NN_p(\cdot)$ denotes the p nearest neighbors. The maximization problem (II) can be converted to solve a generalized eigenvalue problem as follow:

$$(UHU^T)A = \lambda(XLX^T)A \quad (4)$$

where $H = E - B$, $L = D - W$ are Laplacian matrices, E and D are diagonal matrices with their diagonal elements being the column or row(B and W are symmetric) sums of B and W , respectively.

2.2 Derivation of KLPDA

Although LPDA is successful in many circumstances, it often fails to deliver good performance when face images are subject to complex nonlinear changes due to large poses, expressions, or illumination variations, for it is a linear method in nature. In this section, we extend LPDA to its kernel formulation which is to yield a nonlinear locality preserving discriminant subspace by combining the kernel trick and LPDA. The image samples are primarily projected into an implicit high-dimensional feature space, in which different classes are supposed to be linearly separable, by a nonlinear mapping, $\phi : x \in \mathbb{R}^N \mapsto f \in F$. Then the LPDA is conducted in the high-dimensional feature space F . Benefit from the Mercer kernel function, it is unnecessary to compute ϕ explicitly but compute the inner product of two vectors in F with an inner product kernel function: $k(x, y) = \langle \phi(x), \phi(y) \rangle$.

For the same given dataset as in section 2.1, let X^ϕ and U^ϕ be the projections of X and U in F , $y_i = A^T x_i^\phi$, $m_i = A^T u_i^\phi$ be the representations of x_i^ϕ and u_i^ϕ with linear transform A . Define the weight matrices in kernel space $W^\phi = \{diag([w_{ij}^{\phi k}]_{i,j=1}^{n_k})\}_{k=1}^C$ and $B^\phi = [b_{ij}^\phi]_{i,j=1}^C$ in the similar manner with those in the input space

$$w_{ij}^{\phi k} = \begin{cases} \exp(-\frac{\|x_i^\phi - x_j^\phi\|^2}{2t^2}), & x_i^\phi \in NN_p(x_j^\phi) \text{ or } x_j^\phi \in NN_p(x_i^\phi), x_i^\phi, x_j^\phi \in \omega_k. \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

$$b_{ij}^\phi = \begin{cases} \exp(-\frac{\|u_i^\phi - u_j^\phi\|^2}{2t^2}), & u_i^\phi \in NN_p(u_j^\phi) \text{ or } u_j^\phi \in NN_p(u_i^\phi). \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

Unfortunately, we don't know the explicit form of x_i^ϕ and u_i^ϕ in kernel space, it is therefore impossible to compute W^ϕ and B^ϕ directly. Hence, the so-called distance kernel trick is employed to solve this problem, which makes the distances of vectors in kernel space be a function of the distance of input vectors, i.e.,

$$\begin{aligned} \|x_i^\phi - x_j^\phi\|^2 &= \langle x_i^\phi - x_j^\phi, x_i^\phi - x_j^\phi \rangle \\ &= \langle x_i^\phi, x_i^\phi \rangle - 2\langle x_i^\phi, x_j^\phi \rangle + \langle x_j^\phi, x_j^\phi \rangle \\ &= k(x_i, x_i) - 2k(x_i, x_j) + k(x_j, x_j) \end{aligned} \tag{7}$$

Define kernel matrices, $K^{XX} = [k(x_i, x_j)]_{i,j=1}^n$, $K^{UU} = [k(u_i, u_j)]_{i,j=1}^C$, and $K^{XU} = [k(x_i, u_j)]_{i=1, j=1}^{n \times C}$, expression (5) and (6) can be rewritten as

$$w_{ij}^{\phi k} = \begin{cases} \exp\left(-\frac{K_{ii}^{XX} - 2K_{ij}^{XX} + K_{jj}^{XX}}{2t^2}\right), & x_i^\phi \in NN_k(x_j^\phi) \text{ or } x_j^\phi \in NN_k(x_i^\phi), \\ & x_i^\phi, x_j^\phi \in \omega_k. \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

$$b_{ij}^{\phi} = \begin{cases} \exp\left(-\frac{K_{ii}^{UU} - 2K_{ij}^{UU} + K_{jj}^{UU}}{2t^2}\right), & u_i^\phi \in NN_k(u_j^\phi) \text{ or } u_j^\phi \in NN_k(u_i^\phi). \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Then, KLPDA is to maximize the following function

$$J(A) = \frac{\sum_{i,j=1}^c (m_i - m_j) b_{ij}^{\phi} (m_i - m_j)^T}{\sum_{k=1}^c \sum_{\substack{y_i, y_j \in \omega_k \\ y_i \neq y_j}} (y_i - y_j) w_{ij}^{\phi k} (y_i - y_j)^T} = \frac{A^T U^\phi H^\phi (U^\phi)^T A}{A^T X^\phi L^\phi (X^\phi)^T A} \tag{10}$$

where E^ϕ and D^ϕ are diagonal matrices with the diagonal entries being the column or row (B^ϕ and W^ϕ are symmetric) sums of B^ϕ and W^ϕ , $H^\phi = E^\phi - B^\phi$ and $L^\phi = D^\phi - W^\phi$ are Laplacian matrices.

Since any solution of (10), $a_i \in F$, must lie in the span of all the samples in F , there exist coefficients $\psi_i = \{\psi_{ij}\}_{j=1}^n$, such that $a_i = \sum_{j=1}^n \psi_{ij} x_j^\phi = X^\phi \psi_i$, that is $A = X^\phi \Psi$. Thus, problem (10) can be converted to

$$J(A) = \frac{\Psi^T (X^\phi)^T U^\phi H^\phi (U^\phi)^T X^\phi \Psi}{\Psi^T (X^\phi)^T X^\phi L^\phi (X^\phi)^T X^\phi \Psi} = \frac{\Psi^T K^{XU} H^\phi (K^{XU})^T \Psi}{\Psi^T K^{XX} L^\phi (K^{XX})^T \Psi} \tag{11}$$

For convenience, we call $S_b^\phi = K^{XU} H^\phi (K^{XU})^T$, $S_w^\phi = K^{XX} L^\phi (K^{XX})^T$, and $S_t = S_b + S_w$ the kernel locality preserving between-class, within-class, and total scatter matrix respectively. So the problem of (11) is converted to solve the following generalized eigenvalue problem

$$S_b^\phi \Psi = \lambda S_w^\phi \Psi \tag{12}$$

The solution of (12) is consist by the d leading eigenvectors of $(S_w^\phi)^{-1} S_b^\phi$.

3 Regularized KLPDA (RKLPDA)

Obviously, $rank(S_w^\phi) \leq rank(K^{XX}) \leq n$. If S_w^ϕ is full rank, i.e., $rank(S_w^\phi) = n$, then S_w^ϕ is nonsingular and there will be no singularity problem when the matrix $(S_w^\phi)^{-1}S_b^\phi$ is computed. Otherwise, if $rank(S_w^\phi) < n$, where this is always true in face recognition, the SSS problem will occur. For this case, eigenvalue regularization (ER) scheme proposed in [6] is employed to S_w^ϕ . First, perform the eigenvalue decomposition of S_w^ϕ , $S_w^\phi = \Phi_w \Lambda_w \Phi_w^T$, where $\Phi_w = \{\varphi_i^w\}_{i=1}^n$ is the eigenvectors of S_w^ϕ corresponding to the eigenvalues $\Lambda = \{diag(\lambda_i^w)\}_{i=1}^n$, $\lambda_1^w \geq \dots \geq \lambda_r^w \geq \lambda_{r+1}^w = \dots = 0$, r is the rank of the S_w^ϕ . As ER scheme, the eigenspace of S_w^ϕ is decomposed into reliable face space $FS = \{\varphi_k^w\}_{k=1}^m$, unstable noise space $NS = \{\varphi_k^w\}_{k=m+1}^r$, and null space $\emptyset = \{\varphi_k^w\}_{k=r+1}^n$. The starting point of noise region m is set by $\lambda_{m-1}^w = \max\{\forall \lambda_k^w | \lambda_k^w < (\lambda_{med}^w) + \mu(\lambda_{med}^w - \lambda_r^w)\}$, where $\lambda_{med}^w = \text{median}\{\forall \lambda_k^w | k \leq r\}$ is the point near the center of the noise region, μ is a constant, in all experiments of this paper μ is fixed to be 1 for simple. Utilizing the spectrum model $\hat{\lambda}_k^w = \alpha/(k + \beta)$, the eigenvalues are predicted as

$$\hat{\lambda}_k^w = \begin{cases} \lambda_k^w, & k \leq m(\text{facespace}) \\ \alpha/(k + \beta), & m < k \leq r(\text{noisespace}) \\ \alpha/(r + 1 + \beta), & r < k \leq n(\text{nullspace}) \end{cases} \quad (13)$$

where the parameters α and β are given by letting $\hat{\lambda}_1^w = \lambda_1^w$ and $\hat{\lambda}_m^w = \lambda_m^w$. Then using the predicted eigenvalues to weight the corresponding eigenvectors, it has

$$\tilde{\Phi}_w = \{\varphi_k^w / \sqrt{\hat{\lambda}_k^w}\}_{k=1}^n \quad (14)$$

To obtain more features, S_t^ϕ is adopted instead of S_b^ϕ in discriminant feature extraction, since only no more than $C - 1$ features will be obtained when utilizing S_b^ϕ , while $n - 1$ features might be obtained when utilizing S_t^ϕ . The projection of S_t^ϕ in the space spanned by the regularized eigenvectors is

$$\tilde{S}_t^\phi = (\tilde{\Phi}_w)^T S_t^\phi \tilde{\Phi}_w \quad (15)$$

The transformation matrix is consisted of the d leading eigenvectors of \tilde{S}_t^ϕ : $\Phi_t = \{\varphi_i^t\}_{i=1}^d$.

Therefore, for a face image vector x , $x \in \mathfrak{R}^{N \times 1}$, let its projection in kernel space be x^ϕ , then the discriminant features by the proposed RKLPDA method is given by

$$y = A^T x^\phi = (X^\phi \Psi)^T x^\phi = (X^\phi \tilde{\Phi}_w \Phi_t)^T x^\phi = \Phi_t^T \tilde{\Phi}_w^T (X^\phi)^T x^\phi = \Phi_t^T \tilde{\Phi}_w^T K^{Xx} \quad (16)$$

where $K^{Xx} = \{k(x_i, x)\}_{i=1}^n$ is kernel function.

4 Experiments and Discussions

4.1 Face Databases and Image Preprocessing

In all experiments reported in this work, images are preprocessed following the CSU Face Identification Evaluation System [11]. The ORL face database [12]



Fig. 1. Preprocessed sample images of the two databases: (a)ORL database; (b)FERET database

and FERET face database [13] are used for testing. The ORL database contains 400 images from 40 individuals. The FERET database contains 14126 images from 1199 individuals. From FERET database, a subset containing 1131 frontal images from 229 individuals with at least 4 images per individuals, is selected in this work. All the images are scaled to 32×32 pixels and represented by 1024-dimensional vectors. Fig. 1 shows the preprocessed sample images from the two face databases.

4.2 Recognition Experiments and Discussion

In this section, the recognition performances of the proposed RKLPDA with LDA [1], LPDA [2], KFDA [7] and KCLPP [8] are compared. In experiments, cosine polynomial kernel function is chosen, and the parameters are set the same as [14]. For all algorithms, we randomly select i ($i = 2, 3, 4, 5$ for ORL database and $i = 2, 3$ for FERET subset) images of each individual for training and the remaining images for testing. The nearest-neighbor classifier based on cosine distance metric is used for classification. The recognition results from 20 runs are given in Table 1 and 2. Also, an illustration of the recognition accuracies against the number of features on ORL database for $i = 5$ is given in Fig. 2.

From Table 1 and 2, the proposed RKLPDA method consistently and remarkably outperforms the other 4 methods, which validates the effectiveness of the proposed method. In experiments on ORL, the kernel-based methods

Table 1. Recognition accuracy (%) and corresponding number of features on ORL database

TrNum	LDA	LPDA	KFDA	KCLPP	RKLPDA
2	75.5±2.62(39)	58.1±3.05(30)	78.0±2.47(39)	78.4±2.51(39)	79.1±2.55(39)
3	84.5±2.44(39)	77.5±2.10(35)	86.7±2.16(39)	86.0±2.79(39)	89.0±2.29(39)
4	90.5±2.10(39)	87.2±2.49(35)	91.9±1.90(39)	89.5±2.07(39)	94.2±1.62(39)
5	91.9±1.98(39)	91.4±1.85(35)	93.7±1.91(39)	92.6±1.59(39)	96.1±1.08(60)

Table 2. Recognition accuracy (%) and corresponding number of features on FERET database

TrNum	LDA	LPDA	KFDA	KCLPP	RKLPDA
2	68.5±1.74(30)	68.2±1.69(30)	68.4±1.64(228)	40.5±1.77(228)	74.3±1.68(80)
3	79.4±1.34(20)	79.4±1.51(20)	79.0±1.88(228)	58.3±1.97(457)	85.2±1.31(90)

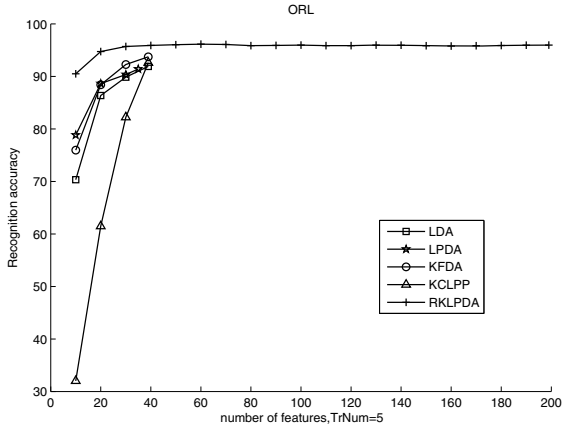


Fig. 2. Recognition accuracy (%) of different algorithms on ORL database

(KFDA and KCLPP) slightly outperform the corresponding linear methods (LDA and LPDA), while in experiments on FERET, the performance of KCLPP is dissatisfactory. This might be caused by the following two reasons: (1) Improper kernel and kernel parameters are chosen; (2) Though KCLPP utilizes the label information, it is not a discriminant analysis method in nature. In addition, as shown in Fig. 2, the LDA, LPDA, KFDA and KCLPP methods obtain at most $C - 1$ discriminant features, while RKLPDA can obtain at most $n - 1$ discriminant features. Meanwhile, the recognition accuracies increase with the increasing of number of samples, and RKLPDA methods achieves a relative good and stable performance with a smaller number of features.

5 Conclusions

This paper presents a regularized kernel locality preserving discriminant analysis (RKLPDA) method. Kernel trick is introduced to extend LPDA to its kernel formulation. To address the singularity problem of kernel locality preserving within-class scatter matrix and utilize the discriminative information in both the principal and null subspace of kernel locality preserving within-class scatter matrix, the eigenvectors are regularized according to the predicted eigenvalues, which de-emphasizes the eigenvectors susceptible to samples noises by raising the eigenvalues, and heavily emphasizes the null space which contains abundant of discriminative information. Extensive experiments on ORL database and FERET subset show that RKLPDA consistently outperforms other linear and kernel methods, which indicates the effectiveness of the proposed method. However, the performance of kernel-based methods diversifies with different kernel functions and kernel parameters, so more attentions on kernel function and kernel parameter choosing should be paid in the future work. Also, the regularization strategy is a key point to be considered.

Acknowledgements. This work is supported by National High-Tech Research and Development Plan of China under Grant no. 2007AA01Z423, the Natural Science Foundation Key Project of CQ CSTC of China under Grant no. CSTC2009AB0175 and the Application Innovation Project of Ministry of Public Security under Grand no. 2010YYCXCQSJ074. The authors would like to thank the anonymous reviewers for their constructive advice.


References

1. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.* 19(7), 711–720 (1997)
2. Yang, L., Gong, W., Gu, X.: Bagging null space locality preserving discriminant classifiers for face recognition. *Pattern Recognit.* 42(9), 1853–1858 (2009)
3. Chen, L., Liao, H., Ko, M.: A new LDA-based face recognition system which can solve the small sample size problem. *Pattern Recognit.* 33(10), 1713–1726 (2000)
4. Yang, L., Gong, W., Gu, X.: Null space discriminant locality preserving projections for face recognition. *Neurocomputing* 71(16–18), 3644–3649 (2008)
5. He, X., Niyogi, P.: Locality preserving projections. In: *Advances in Neural Information Processing System*, vol. 16, pp. 153–160 (2004)
6. Jiang, X., Mandal, B., Kot, A.: Eigenfeature regularization and extraction in face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 30(3), 153–160 (2008)
7. Yang, J., Grangi, A., Yang, J.Y.: KPCA plus LDA: A complete kernel fisher discriminant framework for feature extraction and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 27(2), 230–244 (2005)
8. Li, J., Pan, J.: Kernel class-wise locality preserving projection. *Inf. Sci.* 178(7), 1825–1835 (2008)
9. Pekalska, E., Haasdonk, B.: Kernel discriminant analysis for positive definite and indefinite kernels. *IEEE Trans. Pattern Anal. Mach. Intell.* 31(6), 1017–1031 (2009)
10. Yang, M., Ahuja, N., Kriegman, D.: Face recognition using kernel eigenfaces. *Proc. Int. Conf. Image Process.* 1, 36–40 (2000)
11. FERET Data normalization Procedure, <http://www.cs.colostate.edu/evalfacerec/data/normalization.html>
12. Samaria, F., Harter, A.: Parameterization of a stochastic model for human face identification. In: *Proc. Second IEEE Workshop Application of Computer Vision*, pp. 138–142 (1994)
13. Phillips, P.J., Moon, H., Rizvi, S.: The FERET evaluation methodology for face recognition algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.* 22(10), 1090–1104 (2000)
14. Liu, Q., Lu, H., Ma, S.: Improving kernel fisher discriminant analysis for face recognition. *IEEE Trans. Circuits Syst. Video Technol.* 14(1), 42–49 (2004)

An Appearance-Based Prior for Hand Tracking


Mathias Kölsch

The MOVES Institute, Naval Postgraduate School, Monterey, CA

Abstract. Reliable hand detection and tracking in passive 2D video still remains a challenge. Yet the consumer market for gesture-based interaction is expanding rapidly and surveillance systems that can deduce fine-grained human activities involving hand and arm postures are in high demand. In this paper, we present a hand tracking method that does not require reliable detection. We built it on top of “Flocks of Features” which combines grey-level optical flow, a “flocking” constraint, and a learned foreground color distribution. By adding probabilistic (instead of binary classified) detections based on grey-level appearance as an additional image cue, we show improved tracking performance despite rapid hand movements and posture changes. This helps overcome tracking difficulties in texture-rich and skin-colored environments, improving performance on a 10-minute collection of video clips from 75% to 86% (see examples on our website) 

1 Introduction

While reliable and fast methods to detect and track rigid objects such as faces and cars have matured in the last decade, articulated objects—such as the human body and hand—continue to pose difficult problems to recognition algorithms. The consumer demand for gesture-based interaction, exemplified by the success of and anticipation for the game platforms Wii and Project Natal, has brought about sensing modalities other than color video, including acquisition of depth through active sensors. These are more expensive and less prevalent than video cameras. Particularly, human activity monitoring for elderly care and surveillance applications has to rely on legacy sensors.

Articulated objects present such a difficult challenge because almost every aspect of their characteristics can change: their orientation, size, and shape (silhouette), their apparent color, and their appearance especially due to self-occlusion. No one image cue can be expected to contain sufficient information for detection or tracking. Hence, our approach to overcome these difficulties is to combine many image cues into a rich feature vector that permits more reliable, multimodal hand tracking. We started with a multi-cue method called “Flocks of Features”  (FoF) that combines grey-level LK-feature tracking, a proximity constraint on the tracked features, and a learned foreground color distribution. It can track fast movements and posture changes despite a dynamic background. Still, tracking is difficult if the hand undergoes posture changes *and* the background color is similar to the tracked object’s color *and* the background contains high gradients to which the LK-features might attach. The hand’s appearance—that is, all or part of its grey-level texture—should be taken into consideration for tracking as well. In this paper, we present a method that allows for fast calculation of a

¹ <http://www.movesinstitute.org/%7Ekolsch/paper241Video.wmv>

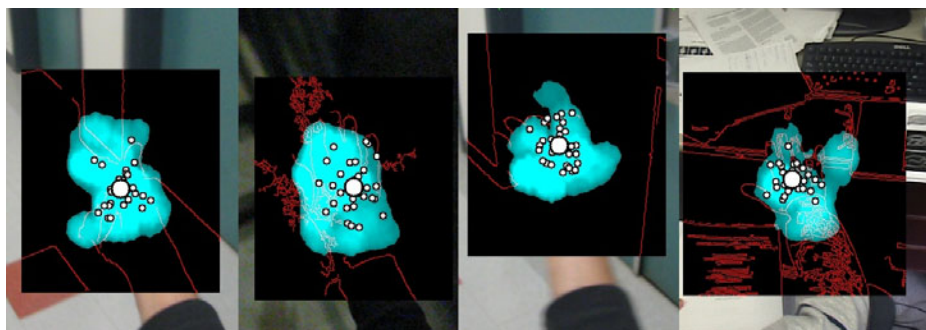


Fig. 1. The appearance-based prior for select hand images

probability that an area’s appearance could be attributed to a hand. Features that have strayed from the flock are then moved to areas of high color *and* appearance probability. While a traditional FoF is agnostic of the object it is tracking except for color and motion consistency, this improved FoF has knowledge of the object’s appearance.

To obtain this appearance-based probability, we trained a Viola-Jones-based detector [20] (VJ) on hands in arbitrary postures and then attempt hand detections at similar scales as the tracked object. Yet, this achieves only poor performance: hands are too varied in appearance for reliable detection. The main innovation of this paper is a method that utilizes *incomplete* detections to make predictions about the presence of a hand. Incomplete detections are areas that successfully passed some but not all VJ cascades. Scores obtained from incomplete detections are integrated over scale and space to yield a prior probability per pixel (see Fig. 1). This image cue is largely orthogonal to color and optical flow, hence providing new information onto which the tracking decision can be based.

The paper is organized as follows. We first discuss the background against which this research has been conducted, including related work. We then present the method to calculate the prior in detail and explain how it is built into FoF tracking. The following experiment section describes the test data and evaluation method, before we present and discuss the results in the last two sections.

2 Background

We briefly discuss related work on object tracking, the traditional Flock of Features (FoF) approach and methods for incomplete detections, or object priors.

2.1 Object Tracking

Rigid objects with a known shape can be tracked reliably before arbitrary backgrounds in grey-level images [17]. However, when the object’s shape varies vastly such as with gesturing hands, most approaches resort to shape-free color information or background differencing [4,9,15]. Yet these approaches rely, for example, on a stationary camera and are not robust to related *unimodal* failure modes. *Multi*-cue methods, on the other

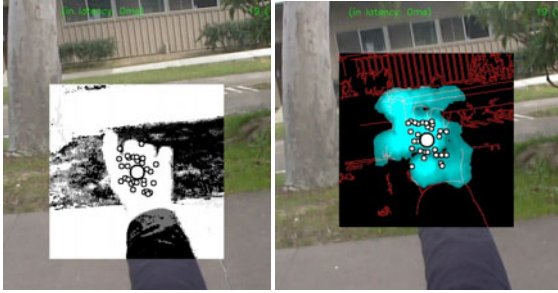


Fig. 2. The color segmentation is useless whereas the appearance probability correctly picks out the hand in front of the walkway

hand, integrate two or more modalities such as texture and color information, allowing recognition and tracking of fixed shapes despite arbitrary backgrounds [3]. The FoF method also employs a multimodal technique to track hands even while changing shape. Here, we add a third modality to further improve tracking.

Pyramid-based LK feature tracking [11][16] describes how to find and track small image artifacts from frame to frame. If the feature match correlation between two consecutive frames is below a threshold, the feature is considered “lost.” (FoF makes use of this feature tracker to track objects, which are larger than, and a composition of, features.)

2.2 Flock of Features Tracking

Flock of Features tracking [8] (FoF) is motivated by the seemingly chaotic flight behavior of a flock of birds such as pigeons. No single bird sets the flock’s direction and the birds frequently exchange relative positions in the flock. Yet, the entire flock stays tightly together as a large “cloud” and is able to perform quick maneuvers and direction changes. Reynolds [13] found that this decentralized organization can be modeled with two constraints: birds like to maintain a minimum safe flying distance to the other birds, but desire not to stray too far from the flock.

LK features, tracked over time, exhibit “flight paths” similar to a flock of birds. Individual features attach to arbitrary artifacts of the object, such as the fingers of a hand. They can then move independently along with the artifact, without disturbing most other features and without requiring the explicit updates of model-based approaches, resulting in flexibility, speed, and robustness. FoF features are constrained to stay a minimum distance apart, yet no more than a maximum distance from their median. Features in violation are repositioned to a conforming location that also has a high skin color probability (see Fig. 2), avoiding dense clusters that ignore parts of the object, and avoiding tracking background artifacts by falling back on a second modality. The FoF can be seen in the various figures and the video as clouds of little dots, their mean (and hand location) as the big dot. Note that FoF tracking—with or without our extension—makes no attempt at estimating the articulation of the hand’s digits (fingers) as model-based approaches do (see for example [2][17]).

One of its strengths is also a weakness: FoF does not rely on an object model beyond object color; the myriad of possible hand configurations does not have to be modeled

explicitly. Here, we introduce a probabilistic appearance-based model that helps constrain the feature locations without placing restrictions on the possible hand configurations and without incurring extraneous computational costs.

2.3 Object Priors

Whereas traditional object detection methods make a binary decision about the presence of the object of interest, our goal is to estimate the probability for the object and to delay the classification decision. Also, instead of a decision for rectangular areas, we need to know the probability per pixel. Lastly, a test area implies a hypothesis about the object's scale, yet we would like an estimate irrespective of scale.

In principle, many object detectors are capable of reporting a score instead of a thresholded classification. Take a PCA-based [19] or wavelet-based [12][14] object descriptor, for example: it measures the distance of the observation from the training mean in image- or feature space. A method is particularly suitable for articulated objects if the different appearances are not aggregated and reduced to a mean. Instead, it must be able to learn dissimilar appearances. For describing dissimilar objects, shape as prior probability has been applied successfully to segmentation and tracking, for example, in an application of the powerful level-set methods [5]. However, appearance-based methods are likely to outperform shape-based methods for natural objects. Yet, appearance-based priors are only recently becoming a popular alternative. Most notable are the excellent tracking and segmentation results of Leibe and Schiele et al. [10].

3 Method

Our method makes three improvements to FoF tracking. First, a posture-independent hand detector is applied to the image at multiple scales, reporting unclassified scores for hand presence. Second, a per-pixel hand probability is calculated from these multi-scale scores of image areas. Third, this hand prior is integrated into the FoF tracking as third image cue and observation modality. This section details each of these steps.

3.1 Hand Scores

If hands could be detected reliably in any posture, tracking by detection would be viable. However, since hands are too varied in appearance, we avoid making the binary classification decision and instead obtain a probabilistic score that directly aids tracking. To calculate a score for an image area to contain an object of interest (at a certain scale and the proper position inside the area), we chose to modify Viola and Jones' detection approach [20] because a) it is very fast, permitting real-time image scanning, b) it is inherently based on local image features, benefitting articulated objects (detect the fingers, not the hand), c) its iterative bootstrapping training method is naturally suited to increasing levels of confidence for object presence, and d) we had prior experience with this method. We are currently evaluating other approaches to calculate this score.

The typical VJ cascade is built with AdaBoost [6] training and consists of N stages, each of which is a linear combination of M weak classifiers. Weak classifiers $h_t(x) \in \{0, 1\}$ make their decision based on intensity comparisons between rectangular

image areas. During testing, stage i is passed successfully if the weighted sum exceeds a stage threshold t_i :

$$\beta_i = \sum_{j=1}^{M_i} \alpha_{ij} h_{ij}(x) \geq t_i \quad (1)$$

All components including weak classifiers, weights and thresholds are learned during the training stage.

A detection occurs when an image area passes all N stages. For our method, we also consider *incomplete detections*, that is, when the image area only passes s stages and gets rejected by stage $s + 1$. We calculate a *score* $o^i(x, y)$ for an image area of scale i , centered at pixel (x, y) as follows. A completely successful detection has passed all N stages, and hence is assigned the score $o = s/N = N/N = 1$. A partially successful detection has passed s stages, $s \in \{0, 1, \dots, N - 1\}$, and is assigned the score $\frac{o=(s+k)}{N}$. Without k , the score is proportional to the number of passed stages. To smooth this step function, k is set to the degree of success within a stage, in the range from zero to exclusive one, $k \in [0; 1)$.

Considering only one stage, k is ideally set proportional to the sum of weights below the threshold t_i :

$$k = \frac{\beta_i - \beta_{min}}{t_i - \beta_{min}}, \text{ where } \beta_{min} = \min_A \sum_{j \in A} \alpha_j h_j(x) \quad (2)$$

for any subset A of weights. Note that the weights α_j can be positive or negative and that the minimum achievable sum β_{min} need not be zero. We avoid computing all combinations of weights to find β_{min} and, instead, set it to a fixed value and ensure $k \geq 0$. This has worked well in practice without negative impact on the generated probability image.

3.2 Formal Justification of Prior

For this score to reflect the probability of a hand, care has to be taken during training to provide the AdaBoost algorithm with a representative set of negative *training* images per stage. If this set is too uniform, then the resulting stage will not proportionally dismiss a more diverse set of negative *test* areas. In other words, if the first few stages do not typically discard test areas at the same rate as later stages, then the score obtained from the first few stages will be artificially inflated. We trained a Viola-Jones-based detector on hands in arbitrary postures and varied the negative training set to avoid such artifacts, allowing us to obtain this appearance-based posture-independent score that an area's appearance could be attributed to a hand.

To aid in placing tracked LK-features, it is desirable to know a probability instead of a score, to know this per pixel instead of per scanned area, and to be considerate of areas scanned at the same center but at multiple scales. The next subsection details how the scores obtained from incomplete detections are integrated over scale and space to yield the prior probability.

3.3 Per-Pixel Object Probability

VJ object detectors require that the actual classifier is scanned across the image, testing rectangular areas at increments in x- and y-position. To detect an object at different scales, either the image needs to be down-scaled or the detector upscaled, typically by 10-25%. After scanning, our modified detector returns one “score image” per scale, its resolution equal to the number of area tests in the x- and y-directions.

An object will typically get detected at multiple adjacent positions and frequently also at nearby scales. The traditional VJ detector heuristically post-processes these detections to combine them into one. Similarly, we devised a way to combine incomplete, rather than binary, adjacent detections. This has the effect of outlier removal and emphasis on actual detections. To this end, every score image is smoothed with a Gaussian. Next, a grey-level morphology (dilation) spreads the point-wise detections to cover a slightly larger area. The combination of the Gaussian covariance, the size of the morphological structuring element, and the number of dilation repetitions should roughly correspond to the size of the object of interest (the hand) within the rectangular VJ area. We chose two configurations, one keeping the point detections rather confined (O_t^i), and one “spreading” them out further (O_s^i , see Fig. 3 and Sec. 5). The resulting point-symmetrical spread is appropriate for hands. Other objects, such as pedestrians, likely benefit from a spread pattern in the shape of the object.

Thereafter, every value is squared to put more emphasis on almost-detections and to devalue not-even-close incomplete detections (remember that the score value is between zero and one). The score images are generally no larger than 160x120 pixels, hence, these are rather quick operations.

Finally, every score image $O_{s/t}^i$ is upscaled to the size of the original video frame and combined with the score images at all resolutions to yield $P_{s/t}$. Since we have fairly good knowledge of the expected scale of the hand in our application, we can constrain the search to such scales and avoid combining scores from much-too-large and much-too-small scales. The desired operation emphasizes detections at the same location in nearby scales, without penalizing detections only at a single scale. Hence, we add the scores, capping them at one. (A max operator would not emphasize, and multiplication

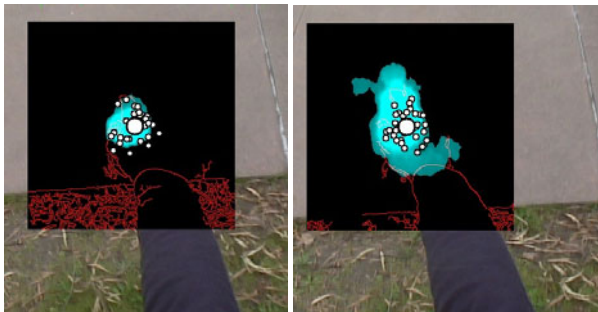


Fig. 3. Small vs. large “spreading” of incomplete detections

would be low-score dominated.) This yields an estimate $P_{s/t}(x, y)$ of a pixel belonging to the hand, irrespective of scale:

$$O_s^i = \gamma_s(D^i \otimes G_{3 \times 3, \sigma} \oplus S_{7 \times 7})^2 \quad (3)$$

$$O_t^i = \gamma_t(D^i \otimes G_{3 \times 3, \sigma} \oplus S_{5 \times 5} \oplus S_{5 \times 5})^4 \quad (4)$$

$$P_{s/t} = \min(1, \sum_i \uparrow O_{s/t}^i) \quad (5)$$

D^i are the incomplete detections at scale i (skipping the image coordinates (x, y)), S is an elliptical structuring element for dilation (\oplus), G is the Gaussian, γ a constant factor, and \uparrow is the upscale operator.

3.4 Multimodal Integration

The hand appearance probability calculated as described above, together with the grey-level optical flow with flocking constraint from the feature tracking, and the particular hand color learned at initial hand detection make for three largely orthogonal image cues that need to be combined into one tracking result. We first combine the color and appearance cues into a joint probability map which is then used to aid the feature tracking.

Preliminary experiments with the joint probability of color and appearance (using their minimum, maximum, weighted average, and product) found that treating the two probabilities as statistically independent distributions and multiplying them yielded the best results: $P_{hand} = P_{color}P_{s/t}$.

For fusion with the tracking information, we follow the same approach as with the original FoF. If a feature is “lost” between frames because the image mark it tracked disappeared or if it violated the flocking constraints, it is moved to a random area of high appearance color probability ($p > 0.5$). If this is not possible without repeated violation of the flocking conditions, it is chosen randomly. Hence, this improved FoF can take advantage of the object’s appearance by relocating features to pixels that “look like hand.” The result is an improvement to the feature re-localization method as the previous approach could not distinguish between the object of interest and background artifacts.

As with the original FoF, this method leads to a natural multimodal integration, combining cues from feature movement based on grey-level image texture with cues from texture-less skin color probability and object-specific texture. Their relative contribution is determined by the desired match quality for features between frames. If this threshold is low, features are relocated more frequently, raising the importance of the color and appearance modalities, and vice versa.

4 Experiments

We compared the performance of the traditional FoF tracking to two parameterizations of FoF with appearance-based prior. We also investigated whether the appearance cue could replace the color cue entirely. The features and color information were initialized

in the same fashion for all configurations, through automatic detection of an “initialization” posture (see [8]). We did not compare against the CamShift tracker [2] since superior performance of traditional FoF tracking was shown already [8].

4.1 Video Sequences

We recorded a total of 16,042 frames of video footage in 13 sequences, over 10 minutes in total, including five of the sequences from [8]. Each sequence follows the motions of the right hand of one of three people, some filmed from the performer’s point of view, some from an observer’s point of view. The videos were shot in an office, a lab, and a hallway as well as at various outdoor locations in front of walkways, vegetation, walls and other common scenes. The videos were recorded with a hand-held DV camcorder, a webcam, and a digital still camera in video mode, then copied to our computer and processed in real-time. A sample video (excerpts from sequence 12) is available from our web site² showing FoF tracking (big and little dots), the color model backprojection (in white) and the appearance prior. The appearance-based probability is shown in cyan, overlaid over a red edge image to improve viewing. (The edges were not used for any calculation.)

5 Results

Following the FoF evaluation [8], we consider tracking to be lost when the mean location (the big dot) is no longer on the hand. The wrist is not considered part of the hand. The tracking for the sequence was stopped then, even though the hand might coincidentally “catch” the tracked feature points again. Since the average feature location cannot be guaranteed to be on the center of the hand or any other particular part, measuring the distance between the tracked location and some ground truth data is not an accurate measure for determining tracking loss. We thus visually inspected and manually annotated every video sequence.

Fig. 4 shows the time until tracking was lost, normalized to the length of the video sequence. The rightmost bars are the average over all sequences. The appearance-added FoF (with larger spread, see below) tracks the hand on average 13.9% longer than the original FoF. As expected, appearance-based FoF can handle some cases where both the flocking and the color modalities break down. Fig. 2 shows two screen shots from sequence 12 where the hand is in front of a walkway and color segmentation does not yield a good result. The hand appearance, however, is visibly distinct from the background and our method produces a high probability for hand pixels. LK feature tracking fails shortly after, and only re-localization on high appearance probabilities allows the hand tracking to continue successfully.

5.1 Spreading Incomplete Detections

Incomplete detections are post-processed as explained in Sec. 3.3. We experimented with two sets of parameters, shown in Eq. 3 (O_s^i , larger spread) and 4 (O_t^i smaller

² <http://www.movesinstitute.org/%7Ekolsch/paper241Video.wmv>

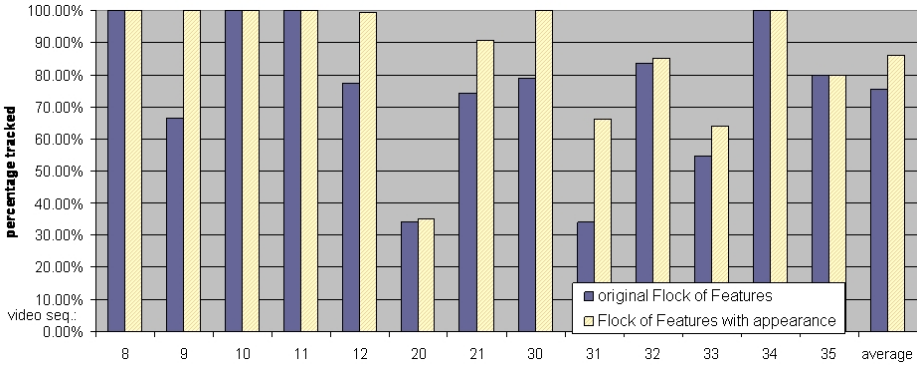


Fig. 4. Time until tracking loss: comparing the original FoF to FoF with appearance cue added, normalized to the length of every video sequence. The rightmost bars are the average over all sequences.

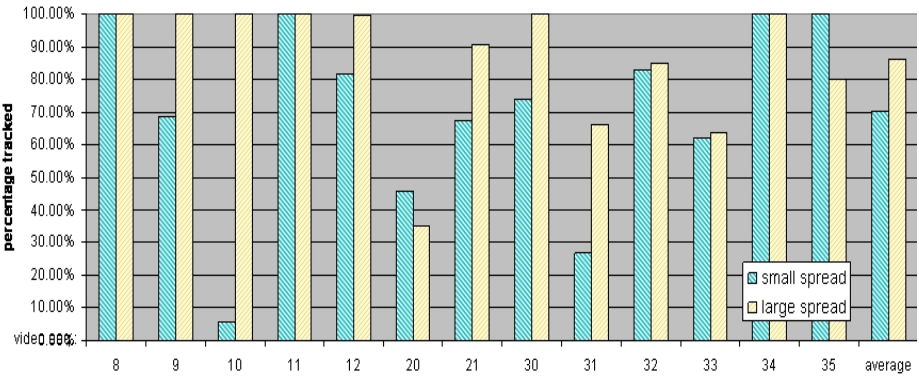


Fig. 5. A wider spread of incomplete detections outperforms too narrow an influence area

spread) to test whether a larger or smaller neighborhood of higher appearance probabilities would yield better results. Fig. 5 shows the tracking length on the same sequences as above and their average.

Spreading incomplete detections only a small amount in fact hurts the performance by, on average, 6.9% (70.48%) compared to the original FoF (75.68%, see Fig. 4). Whereas spreading the incomplete detections to a larger area (Eq. 3) improves tracking by 22.3% over the small spread, or 13.9% over original FoF tracking. Small-spread suffers from poor performance early on in sequence 10. Not counting this sequence, its performance would go up to 77.74%, which is better than original FoF tracking but still not as good as with a wider spread.

5.2 Appearance versus Color

One might consider replacing the color cue entirely with the appearance-based probability. However, as Fig. 6 illustrates, the performance suffers significantly, whether with

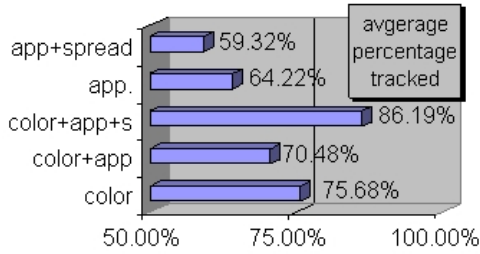


Fig. 6. Contribution of appearance versus color. Shown are the averages over all sequences.

or without a larger “spread.” One notes also that spreading incomplete detections without combining with the color probability exhibits even worse performance, and only in the combination into a three-cue versus a two-cue tracker does the appearance-based FoF tracker’s full potential get realized.

6 Discussion

Most of the performance improvement can be attributed to the appearance cue providing useful information when the color segmentation fails and considers nearly every pixel as skin-colored. If LK features are lost or in violation of the flocking constraints during those cases, the appearance cue limits the placement of re-located features to likely hand pixels, instead of landing on background artifacts that happen to be skin-colored. The color modality generally fails if the tracking initialization is poor (no good match between observed hand location and mask), and if extensive camera motion changed the composition of the background color.

A wider spread considers rather more than fewer pixels of hand appearance, due to the imprecise segmentation achieved. Hence, the color cue is often still very important, particularly with very cluttered backgrounds in which the hand detection returns rather high scores. Given these considerations, failures most frequently occur when the hand posture changes in front of skin-colored, cluttered backgrounds.

Appearance on its own is currently an inferior cue to color. If a good color histogram is learned during tracking initialization, it provides an excellent and very precise cue for which pixels belong to the hand and which do not. As articulated object detection improves, we expect the appearance cue to become more important. Equally, segmentation during detection (e.g., [10,18]) can supply probabilistic segmentation with better resolution than Eq. 3.5 in turn improving the value of the appearance cue.

Traditional FoF tracking favors objects with more distinct and more uniform color. Tracking with this extension to FoF excels in performance if the object has a distinct appearance.

7 Conclusions

The power of the traditional FoF tracking lies in its combination of two image cues so that it can continue to track successfully even if only one “constancy” assumption

is violated. Our improved approach integrates three image cues, making it robust to incorrect information from two modalities.

We demonstrated improved performance with an appearance-based prior on the tracking of hands in videos without a static background. As an extension to Flock of Features tracking it provides a third, orthogonal image cue that the tracking decision can be based on. This makes it more robust to strong background gradients, background in a color similar to the hand color, and rapid posture changes. The resulting tracking method is rather robust and operates indoors and outdoors, with different people, and despite dynamic backgrounds and camera motion. The method was developed for a real-time gesture recognition application and currently requires around 20ms per 720x480 video frame.

Despite the advantages of the chosen appearance-based prior, we are currently evaluating the performance of other methods including some for whole objects and some parts-based approaches [18,10]. The integration of their results into this tracking framework follows the same multimodal fusion approach as this paper's contribution.

While the current interest in virtual and augmented reality as well as 3D technologies provides ample applications and need for good hand tracking, this method is not limited to hands but likely also applicable to tracking of other articulated objects such as pedestrians, for example, for surveillance applications.

References

1. Birchfield, S.: Elliptical head tracking using intensity gradients and color histograms. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 232–237 (June 1998)
2. Bradski, G.R.: Real-time face and object tracking as a component of a perceptual user interface. In: Proc. IEEE Workshop on Applications of Computer Vision, pp. 214–219 (1998)
3. Bretzner, L., Laptev, I., Lindeberg, T.: Hand Gesture Recognition using Multi-Scale Colour Features, Hierarchical Models and Particle Filtering. In: Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition, Washington D.C., pp. 423–428 (2002), <http://citeseer.nj.nec.com/bretzner02hand.html>
4. Cutler, R., Turk, M.: View-based Interpretation of Real-time Optical Flow for Gesture Recognition. In: Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition, pp. 416–421 (April 1998)
5. Dambreville, S., Rathi, Y., Tannenbaum, A.: A Framework for Image Segmentation Using Shape Models and Kernel Space Shape Priors. IEEE Trans. Pattern Analysis and Machine Intelligence 30(8) (August 2008)
6. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of online learning and an application to boosting. In: Vitányi, P.M.B. (ed.) EuroCOLT 1995. LNCS, vol. 904, pp. 23–37. Springer, Heidelberg (1995)
7. Isard, M., Blake, A.: A mixed-state CONDENSATION tracker with automatic model-switching. In: Proc. Intl. Conference on Computer Vision, pp. 107–112 (1998)
8. Kölsch, M., Turk, M.: Fast 2D Hand Tracking with Flocks of Features and Multi-Cue Integration. In: IEEE Workshop on Real-Time Vision for Human-Computer Interaction, CVPR (2004), <http://www.cs.ucsb.edu/~7Ematz/HGI/KolschTurk2004Fast2DHandTrackingWithFlocksOfFeatures.pdf>

9. Kurata, T., Okuma, T., Kourogi, M., Sakaue, K.: The Hand Mouse: GMM Hand-color Classification and Mean Shift Tracking. In: Intl. Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-time Systems (July 2001), <http://www.is.aist.go.jp/vizwear/Demo-e/Handmousee.html>
10. Leibe, B., Leonardis, A., Schiele, B.: Robust Object Detection with Interleaved Categorization and segmentation. *Int. Journal of Computer Vision* 77(1), 259–289 (2008)
11. Lucas, B.D., Kanade, T.: An Iterative Image Registration Technique with an Application to Stereo Vision. In: Proc. Imaging Understanding Workshop, pp. 121–130 (1981)
12. Oren, M., Papageorgiou, C., Sinha, P., Osuna, E., Poggio, T.: Pedestrian Detection Using Wavelet Templates. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (June 1997)
13. Reynolds, C.W.: Flocks, Herds, and Schools: A Distributed Behavioral Model. *ACM Trans. on Graphics (Proc. SIGGRAPH)* 21(4), 25–34 (1987)
14. Schmidt-Feris, R., Gemmell, J., Toyama, K., Krüger, V.: Hierarchical Wavelet Networks for Facial Feature Localization. In: Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition (May 2002), <http://research.microsoft.com/~JGemmell/pubs/FerisFG2002.pdf>
15. Segen, J., Kumar, S.: GestureVR: Vision-Based 3D Hand Interface for Spatial Interaction. In: Proc. ACM Intl. Multimedia Conference (September 1998), http://www.acm.org/sigs/sigmm/MM98/electronic_proceedings/segen/
16. Shi, J., Tomasi, C.: Good features to track. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Seattle (June 1994), citeseer.nj.nec.com/shi94good.html
17. Stenger, B., Thayananthan, A., Torr, P.H.S., Cipolla, R.: Filtering Using a Tree-Based Estimator. In: Proc. 9th International Conference on Computer Vision, Nice, France, vol. II, pp. 1063–1070 (October 2003)
18. Torralba, A., Murphy, K.P., Freeman, W.T.: Sharing Visual Features for Multiclass and Multiview Object Detection. *IEEE Trans. Pattern Analysis and Machine Intelligence* 29(5), 854–869 (2007)
19. Turk, M., Pentland, A.: Eigenfaces for Recognition. *J. Cognitive Neuroscience* 3(1), 71–86 (1991)
20. Viola, P., Jones, M.: Robust Real-time Object Detection. In: Intl. Workshop on Statistical and Computational Theories of Vision (July 2001), <http://citeseer.nj.nec.com/viola01robust.html>
21. Wu, Y., Huang, T.S.: Hand Modeling, Analysis, and Recognition. *IEEE Signal Processing Magazine* (May 2001)

Image Recognition through Incremental Discriminative Common Vectors*

Katerine Díaz-Chito, Francesc J. Ferri, and Wladimiro Díaz-Villanueva

Dept. Informàtica, Universitat de València. Spain
{Katerine.Diaz,Francesc.Ferri,Wladimiro.Diaz}@uv.es

Abstract. An incremental approach to the discriminative common vector (DCV) method for image recognition is presented. Two different but equivalent ways of computing both common vectors and corresponding subspace projections have been considered in the particular context in which new training data becomes available and learned subspaces may need continuous updating. The two algorithms are based on either scatter matrix eigendecomposition or difference subspace orthonormalization as with the original DCV method. The proposed incremental methods keep the same good properties than the original one but with a dramatic decrease in computational burden when used in this kind of dynamic scenario. Extensive experimentation assessing the properties of the proposed algorithms using several publicly available image databases has been carried out.

1 Introduction

Representing images in appropriate subspaces in order to dramatically reduce the volume of the corresponding data to improve their discriminability is a common trend in many image recognition algorithms proposed to date [1,2]. When applied to very large images, these methods imply relatively high time and space requirements as they are usually need non trivial numerical operations on large matrices computed from a previously given training set.

In particular dynamic or interactive scenarios, image recognition algorithms may require retraining as new information becomes available. New (labeled) data may be then added to the previous training set so that the original (batch) algorithm can be used but this involves prohibitive computational burden for most practical applications. Instead, incremental subspace learning algorithms have been proposed for basic algorithms such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) in order to alleviate these requirements while keeping most of the performance properties of its batch counterpart [3,4,5,6,7].

Subspace learning methods based on Discriminative Common Vectors (DCV) have been recently proposed for face recognition [2]. The rationale behind DCV

* Work partially funded by FEDER and Spanish and Valencian Governments through projects TIN2009-14205-C04-03, ACOMP/2010/287, GV/2010/086 and Consolider Ingenio 2010 CSD07-00018.

is close to LDA but is particularly appealing because its good performance behavior and flexibility of implementation specially in the case of very large dimensionalities [8,2].

In this paper, incremental formulations corresponding to basic (batch) implementations of the DCV method are proposed. The derived algorithms follow previously published ideas about (incrementally) modifying subspaces [9,10] but in the particular context of DCV. Both subspace projections and explicit vectors are efficiently recomputed allowing the application of these algorithms in interactive and dynamic problems.

2 Discriminant Common Vectors for Image Characterization and Recognition

The DCV method has been recently proposed for face recognition problems in which input data dimension is much higher than the training set size [2]. In particular, the method looks for a linear projection that maximizes class separability by considering a criterion very similar to the one used for LDA-like algorithms and also uses the within-class scatter matrix, S^w . In short, the method consists of constructing a linear mapping onto the null space of S^w in which all training data gets collapsed into the so-called *discriminant common vectors*. Classification of new data can be then accomplished by first projecting it and then measuring similarity to DCVs of each class with an appropriate distance measure.

Let $\mathcal{X} \in \mathbb{R}^{d \times M}$ be a given training set consisting of M d -dimensional (column) vector-shaped images, $x_j^i \in \mathbb{R}^d$, where $i = 1, \dots, M_j$ refers to images of any of the c given classes, $j = 1, \dots, c$ and $M = \sum_{j=1}^c M_j$. Let S_X^w be their corresponding within-class scatter matrix and let x_j be the j -th class mean vector from \mathcal{X} .

2.1 DCV through Eigendecomposition

Let $U \in \mathbb{R}^{d \times r}$ and $\bar{U} \in \mathbb{R}^{d \times n}$ be matrices formed with the eigenvectors corresponding to non zero and zero eigenvalues, computed from the eigenvalue decomposition (EVD) of S_X^w where r and $n = d - r$ are the dimensions of its range and null spaces, respectively. The j -th class common vector can be computed as the orthonormal projection of the j -th class mean vector onto this null space, $\bar{U}\bar{U}^T x_j$ or, equivalently as the residue of x_j with regard to U . That is

$$x_{com}^j = x_j - UU^T x_j \quad (1)$$

In both expressions, the mean vector x_j may in fact be substituted by any other j -class training vector [2]. Note that it is much easier and convenient to use U rather than \bar{U} , partially because in the context of image recognition usually $r \ll n$.

These d -dimensional common vectors constitute a set of size c to which standard PCA can be applied. The combination of this with the previous mapping gives rise to a linear mapping onto a reduced space, $W \in \mathbb{R}^{d \times (c-1)}$. Reduced

dimensionality discriminative common vectors (DCVs) can be then computed as $\Omega_j = W^T x_j$. When new (test) data, x , is to be classified, it can get projected as $W^T x$ and then appropriately compared to Ω_j in order to be recognized.

Even after several improvements that can be applied [11,2], the computational burden associated to this procedure is dominated by the eigendecomposition of S_X^w and leads to a cost in $O(\ell M^3 + dM^2)$, where ℓ is a constant related to the iterative methods used for EVD.

2.2 DCV through Orthonormalization

An alternative and more efficient way of computing an equivalent projection requires the use of Gram-Schmidt orthonormalization (GSO) instead of EVD.

Let $\mathcal{B}_X \in \mathbb{R}^{d \times (M-c)}$ be a matrix whose columns are given by difference vectors $x_j^i - x_j^1$, where $j = 1, \dots, c$, and $i = 2, \dots, M_j$. It can be shown that the range subspace of S_X^w and the subspace spanned by \mathcal{B}_X are the same. Therefore, a mapping $\Theta \in \mathbb{R}^{d \times r}$ can be computed using the r base vectors obtained from \mathcal{B}_X through GSO. This mapping can equivalently substitute the mapping U in Equation 1 to compute the same common vectors.

The difference common vectors,

$$\mathcal{B}_{com}^X = [(x_{com}^2 - x_{com}^1) \ \dots \ (x_{com}^c - x_{com}^1)] \in \mathbb{R}^{d \times (c-1)},$$

can now be computed and a linear mapping to a reduced space is obtained from \mathcal{B}_{com}^X using GSO. The composition of this mapping with the previous one leads to a linear mapping, Ψ , equivalent (but different in general) to the composite mapping in the previous section, W . This mapping represents the same subspace as W given that $WW^T = \Psi\Psi^T$.

As in the previous case, the cost of obtaining the reduced mapping can be neglected with regard to the cost of computing the projection Θ that amounts to $O(dM^2)$.

3 Incrementally Computing Discriminative Common Vectors

Both basic algorithms in the previous section have a first phase in which projections (U or Θ) are obtained in order to apply Equation 1. And a second one in which a definitive mapping (W or Ψ) is obtained. From an algorithmic viewpoint, the second phase mimics the first one at a much smaller scale in both cases. Consequently, only details about the first phase will be given here.

Let \mathcal{X} be as defined in Section 2 and let $\mathcal{Y} \in \mathbb{R}^{d \times N}$ be the currently available training set (new incoming data) consisting of N_j vectors from each of the c classes. And let $\mathcal{Z} = [\mathcal{X} \ \mathcal{Y}] \in \mathbb{R}^{d \times (M+N)}$.

3.1 Incremental DCV through EVD

Basic DCV method on \mathcal{Z} would require eigendecomposing $S_{\mathcal{Z}}^w$ first in order to use Equation 1 to compute the common vectors and go forward. To avoid this,

S_Z^w must be decomposed into simpler parts that can be put in terms of eigendecompositions $S_X^w = UAU^T$ (from the previous iteration) and $S_Y^w = V\Delta V^T$ (that can be done straightaway as $N \ll M$) along with their corresponding mean vectors x_j and y_j . From the standard within-class scatter matrix definition we can arrive at

$$S_Z^w = S_X^w + S_Y^w + \mathcal{S}\mathcal{S}^T \tag{2}$$

which could be seen as a generalization of the decomposition in [9]. In this expression, $\mathcal{S} \in \mathbb{R}^{d \times c}$ is a matrix whose columns are defined in terms of mean vectors from \mathcal{X} and \mathcal{Y} as $\sqrt{\frac{M_j N_j}{(M_j + N_j)}}(x_j - y_j)$ for each class j .

To effectively arrive at a convenient eigendecomposition of S_Z^w , an orthonormal basis, $[U \ v] \in \mathbb{R}^{d \times s}$ (where s is the rank of S_Z^w), spanning \mathcal{S} and the centered versions of \mathcal{X}, \mathcal{Y} (that is, range spaces of S_X^w and S_Y^w) needs to be obtained. The unknown $v \in \mathbb{R}^{d \times (s-r)}$ is computed by using the residual operator with regard to U (as in Equation 1) of added subspaces (related to \mathcal{Y} and \mathcal{S}) and then applying GSO to the composite residual set $[(V - UU^T V) (\mathcal{S} - UU^T \mathcal{S})]$ (after removing any zero vectors).

As $[U \ v]$ only differs from the sought U' (in $S_Z^w = U' A' U'^T$) in a rotation, R , we can now write

$$S_X^w + S_Y^w + \mathcal{S}\mathcal{S}^T = [Uv]R \ A'R^T[Uv]^T$$

and modify it to have instead:

$$R A' R^T = \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} U^T V \Delta V^T U & U^T V \Delta V^T v \\ v^T V \Delta V^T U & v^T V \Delta V^T v \end{bmatrix} + \begin{bmatrix} U^T \mathcal{S} \mathcal{S}^T U & U^T \mathcal{S} \mathcal{S}^T v \\ v^T \mathcal{S} \mathcal{S}^T U & v^T \mathcal{S} \mathcal{S}^T v \end{bmatrix}$$

which constitutes a new eigenproblem that allows us to compute R and correspondingly $U' = [U \ v]R$.

The above computation needs $O(\ell(N^3 + s^3) + d(N^2 + s^2))$ time and dominates the cost of the whole incremental algorithm that will be referred to as IDCV-EVD. This constitutes an improvement with respect to the corresponding basic algorithm which would imply a computation time in $O(\ell(M + N)^3 + d(M + N)^2)$. Note that the benefit will be higher if the rank of the overall scatter matrix, s , is reduced. This can be easily done by neglecting small eigenvalues in the EVD decompositions used.

3.2 Incremental DCV through GSO

An incremental version of the GSO-based DCV is also possible by constructing $\mathcal{B}_Y \in \mathbb{R}^{d \times N}$, the difference vectors in \mathcal{Y} with regard to the *same* samples as \mathcal{B}_X , namely as $y_j^k - x_j^1$, with $j = 1, \dots, c$, and $k = 1, \dots, N_j$.

In this case, GSO can be applied to \mathcal{B}_Y starting with the orthonormal basis previously computed for \mathcal{B}_X , Θ , to add new vectors to complete an incremented orthonormal basis, Θ' , that spans the whole difference set, $[\mathcal{B}_X \ \mathcal{B}_Y] \in \mathbb{R}^{d \times (M+N-c)}$.

As with IDCV-EVD, the computation of Θ' dominates the cost of the whole algorithm, which will be referred to as IDCV-GSO. In this case, the cost of the GSO-based DCV algorithm can be cut from $O(d(M+N)^2)$ to $O(dN^2)$ if all new samples are linearly independent.

4 Experiments and Discussion

A number of experiments have been carried out to assess the relative benefits of the IDCV algorithms with regard to the direct methods using data in a range of situations. In this work, 3 publicly available image databases have been considered. Images were previously normalized in intensity, scaled and aligned using the position of eyes and mouth. Figure 1 shows some sample images and their basic characteristics as dimensionality (image size), number of classes (c), number of objects (per class), and type of variability. More details about these databases can be found in the corresponding references also given as part of Figure 1.

In particular, an experimental setup in which more training data becomes available to the algorithm has been designed. For each database, the available data has been split into 3 disjoint sets. The first two are test (20%) and initial training set (30%), respectively and the remaining 50% is made available as new training data in portions of N images per class. Starting from a random permutation of the images, test and train blocks in the partition have been shifted throughout all the database so that all images have been used as test after each evaluation round. Moreover, the incremental data subset has been randomly permuted after each shift to remove any kind of dependence on the order in which data is made available to the algorithm. The results presented correspond then to an average across the whole database along with corresponding standard deviations.

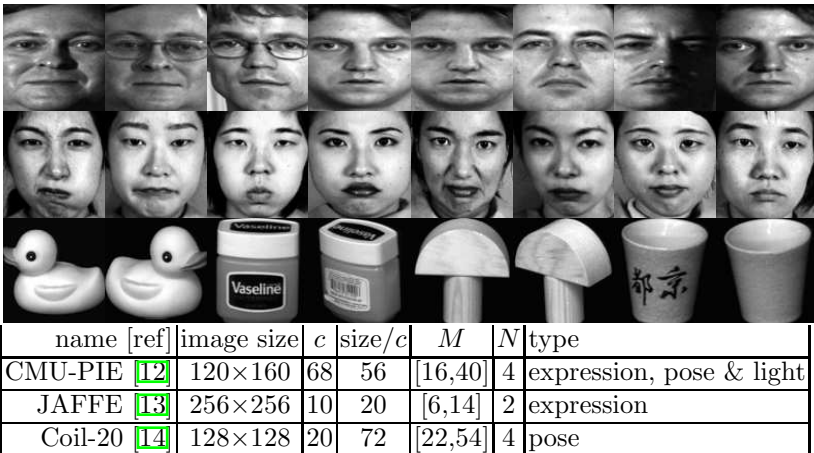


Fig. 1. Sample images from the 3 databases used in the experiments along with corresponding details

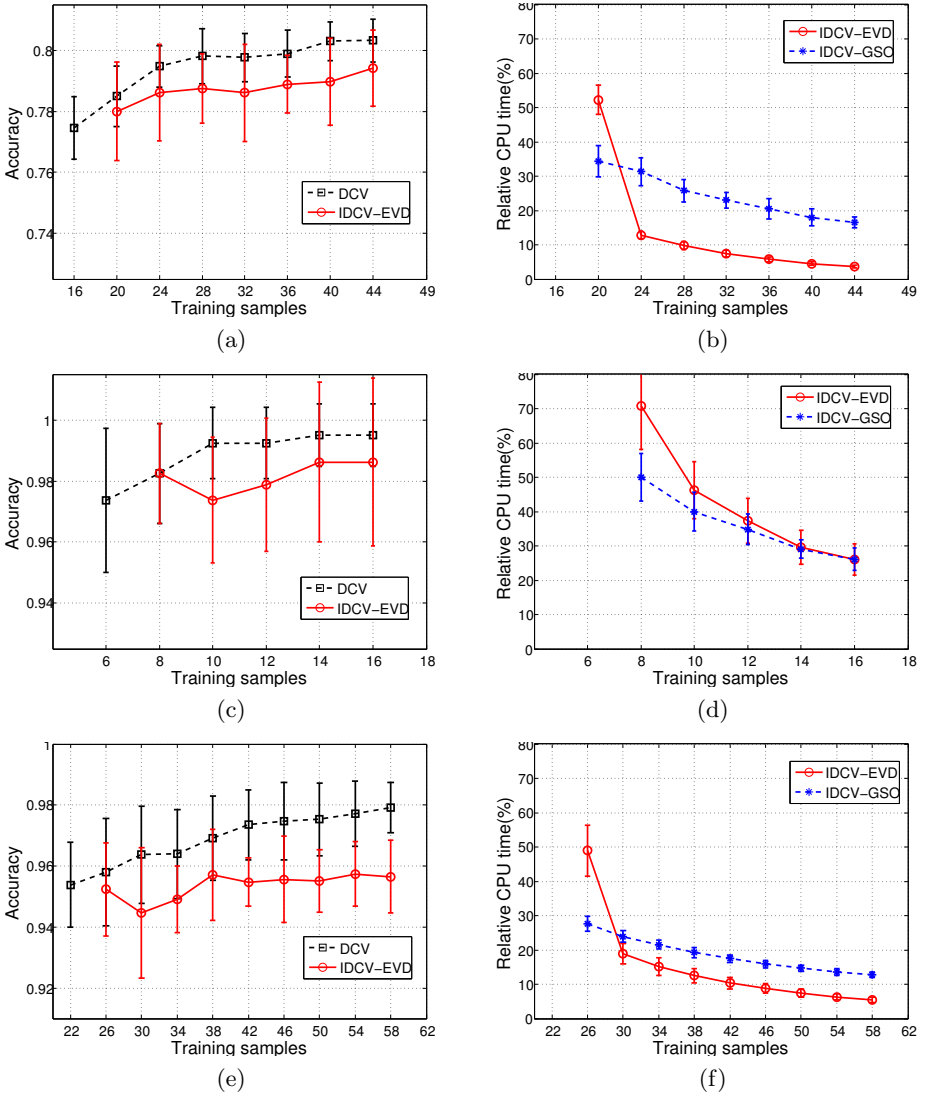


Fig. 2. Results obtained with the 3 databases, CMU-PIE, JAFFE and Coil-20, one at each row. (a)(c)(e) Averaged accuracy vs accumulated training set size for DCV and IDCV-EVD. Results with IDCV-GSO and DCV are identical. (b)(d)(f) Relative CPU time of both IDCV methods with regard to corresponding DCV ones.

At each iteration, N new images per class are available. The IDCV algorithms are then run using the previous M images. The basic DCV algorithm is also run from scratch using the current $M + N$ images. In this way, M values range approximately from 30 to 80% while the value of N has been fixed for each

database according to its global size (in the range from 6 to 10%). Particular M and N values used are also shown in Figure 1.

The accuracy of the minimum distance classifier using DCVs in the projected subspace has been considered [2]. Also, relative CPU time for each incremental algorithm at each iteration with regard to the basic DCV algorithm has been adopted to put forward the relative computational benefit of using incremental versus batch algorithms. Both, classification and efficiency results are shown in Figure 2.

Accuracy plots clearly show that there is not a significant difference between incremental and batch classification results as expected. In the case of IDCV-GSO, classification results are exactly the same as with DCV since this incremental procedure is less prone to numerical errors. On the other hand, a small decrease is observed in all cases when comparing IDCV-EVD to DCV due to numerical inaccuracies when computing eigendecompositions. Take also into account that we could have fixed a more strict tolerance level in the numerical procedures but this would have had an impact in the computation times. It is worth noting that with our current implementation, the observed degradation in performance is kept into an insignificant level as the training set is increased. More interestingly, relative times plot in Figure 2 exhibit relative savings from about 30% up to 95% of the time spent by the basic DCV algorithm. Obviously, the relative CPU time decreases with M while N is kept fixed.

Several interesting facts can be put forward. First, the IDCV-GSO algorithm gets significantly higher savings than the IDCV-EVD one in the first iteration (smallest value of M). This situation is only partially kept for the smallest database. On the contrary, IDCV-EVD is more efficient than IDCV-GSO (with regard to its batch counterpart) for larger values of M . This behavior gets more evident in the case of the largest database.

Both incremental algorithms are able to cut computational cost to 25% or less of their corresponding batch algorithm. Preference to use one or another will depend also on absolute computation times which in turn may depend on the particular implementation. For example, in our unoptimized implementation, GSO is roughly 5 times slower than EVD. With a more careful and efficient implementation this situation could be turned upside-down [2]. Regardless of computational cost, IDCV-EVD may lead to some additional benefits as it permits controlling the size of the null space as in [15] which may help in increasing the generalization ability of the incremental algorithm.

5 Concluding Remarks and Further Work

Incremental algorithms to compute DCVs and corresponding subspaces have been proposed. The algorithms use incremental eigendecomposition and Gram-Schmidt orthonormalization, respectively as in the original (batch) algorithms. Dramatic computational savings are observed while performance behavior of DCV is preserved.

Further work is driven towards the implementation of more general common vector based subspace algorithms, using extended null space and kernels, in an

incremental way along with extending the experimentation to other, more challenging truly dynamic scenarios. In particular, biometric recognition applications with limited resources (i.e. mobile platforms) in which templates of different users may need constant and frequent updates are the target applications.

References

1. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 19(7), 711–720 (1997)
2. Cevikalp, H., Neamtu, M., Wilkes, M., Barkana, A.: Discriminative common vectors for face recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence* 27(1), 4–13 (2005)
3. Murakami, H., Kumar, B.: Efficient calculation of primary images from a set of images. *IEEE Trans. Patt. Analysis and Machine Intell* 4(5), 511–515 (1982)
4. Chandrasekaran, S., Manjunath, B., Wang, Y., Winkler, J., Zhang, H.: An eigenspace update algorithm for image analysis. *Graphical Models and Image Processing* 59(5), 321–332 (1997)
5. Hall, P.M., Marshall, D., Martin, R.R.: Incremental eigenanalysis for classification. In: *British Machine Vision Conference*, pp. 286–295 (1998)
6. Ozawa, S., Toh, S.L., Abe, S., Pang, S., Kasabov, N.: Incremental learning of feature space and classifier for face recognition. *Neural Netw.* 18(5-6), 575–584 (2005)
7. Ross, D.A., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking. *Int. J. Comput. Vision* 77(1-3), 125–141 (2008)
8. Cevikalp, H., Neamtu, M., Wilkes, M., Barkana, A.: A novel method for face recognition. In: *Proceedings of the IEEE 12th Signal Processing and Communications Applications Conference*, pp. 579–582 (2004)
9. Hall, P., Marshall, D., Martin, R.: Merging and splitting eigenspace models. *IEEE Trans on Pattern Analysis and Machine Intelligence* 22(9), 1042–1049 (2000)
10. Hall, P., Marshall, D., Martin, R.: Adding and subtracting eigenspaces with eigenvalue decomposition and singular value decomposition. *Image and Vision Computing* 20(13-14), 1009–1016 (2002)
11. Gulmezoglu, M., Dzhafarov, V., Keskin, M., Barkana, A.: A novel approach to isolated word recognition. *IEEE Trans. Speech and Audio Processing* 7(6), 618–620 (1999)
12. Sim, T., Baker, S., Bsat, M.: The CMU pose, illumination, and expression (PIE) database. In: *Proceedings of the 5th International Conference on Automatic Face and Gesture Recognition* (2002)
13. Lyons, M.J., Akamatsu, S., Kamachi, M., Gyoba, J.: Coding facial expressions with gabor wavelets. In: *Third IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 200–205 (1998)
14. Nene, S., Nayar, S.K., Murase, H.: Columbia object image library (coil-20). Technical report (1996)
15. Tamura, A., Zhao, Q.: Rough common vector: A new approach to face recognition. In: *IEEE Intl. Conf. on Syst, Man and Cybernetics*, pp. 2366–2371 (2007)

Dynamic Facial Expression Recognition Using Boosted Component-Based Spatiotemporal Features and Multi-classifier Fusion

Xiaohua Huang^{1,2}, Guoying Zhao¹, Matti Pietikäinen¹, and Wenming Zheng²

¹ Machine Vision Group, Department of Electrical and Information Engineering,
University of Oulu, Finland

² Research Center for Learning Science, Southeast University, China

{huang.xiaohua, gyzhao, mkp}@ee.oulu.fi,

wenming_zheng@seu.edu.cn

<http://www.ee.oulu.fi/mvg>

Abstract. Feature extraction and representation are critical in facial expression recognition. The facial features can be extracted from either static images or dynamic image sequences. However, static images may not provide as much discriminative information as dynamic image sequences. On the other hand, from the feature extraction point of view, geometric features are often sensitive to the shape and resolution variations, whereas appearance based features may contain redundant information. In this paper, we propose a component-based facial expression recognition method by utilizing the spatiotemporal features extracted from dynamic image sequences, where the spatiotemporal features are extracted from facial areas centered at 38 detected fiducial interest points. Considering that not all features are important to the facial expression recognition, we use the AdaBoost algorithm to select the most discriminative features for expression recognition. Moreover, based on median rule, mean rule, and product rule of the classifier fusion strategy, we also present a framework for multi-classifier fusion to improve the expression classification accuracy. Experimental studies conducted on the Cohn-Kanade database show that our approach that combines both boosted component-based spatiotemporal features and multi-classifier fusion strategy provides a better performance for expression recognition compared with earlier approaches.

Keywords: Component, facial interest point, feature selection, multi-classifier fusion, spatiotemporal features.

1 Introduction

A goal of facial expression recognition is to determine the emotional state, e.g. happiness, sadness, surprise, neutral, anger, fear, and disgust, of human beings based on the facial images, regardless of the identity of the face. To date, most of facial expression recognition are based on static images or dynamic image sequences [1,2,3], where dynamic image sequences based approaches provide more accurate and robust recognition of facial expressions than the static image based approaches.

Facial feature representation in dynamic image sequences is critical to facial expression recognition. Generally, two sorts of features can be extracted, i.e. the geometric features versus the appearance based features. Geometric features, often extracted from the shape and locations of facial components, are concatenated by feature vectors to represent the face geometry. A typical geometric feature extraction procedure can be state as follows: automatically detecting the approximate location of facial feature points in the initial frame, then manually adjusting the points, and finally tracking the changes of all points in the next frame. Most studies focused on how to detect and track motion of facial components based on lips, eyes, brows, cheek through building a geometric model. For example, Tian et al. [4] proposed multi-state models to extract the geometric facial features for detecting and tracking the changes of facial components in near frontal face images. Kobayashi et al. [5] proposed a geometric face model described by 30 facial feature points to this purpose. Appearance features represent texture changes of skin in the face, such as wrinkles and furrows. Some techniques, such as Gabor wavelet representation [6], optical flow [7], independent component analysis (ICA) [8], and local feature analysis (LFA) [9], are widely used to extract the facial appearance features. For example, Kotsia et al. [10] proposed a grid-tracking and deformation system based on deformation models and tracking the grid in consecutive video frames over time. Donato et al. [11] compared the above techniques on analyzing facial actions of the upper and lower face in image sequences. Feng et al. [12] used local binary patterns on small facial regions for describing facial features. However, the major limitation of the geometric features is that they may be sensitive to shape and resolution variations, whereas the appearance features may contain redundant information.

Some researches combine both geometric and appearance features for designing automatic facial expression recognition to overcome the limitation of the geometric and the appearance based features. For example, Lanitis et al. [13] used the active appearance models (AAM) to interpret the face images. Yesin et al. [14] proposed a method to extract positions of the eyes, eyebrows and the mouth, for determining the cheek and forehead regions, and then apply the optical flow on these regions, and finally feed the resulting vertical optical flow values to the discrete Hopfield network for recognizing expressions. Recent studies [15][16] have shown that the combination of geometric and appearance based features can achieve excellent performance in face recognition with robustness to some problems caused by pose motion and partial occlusion. However, these methods are only based on static images, rather than dynamic image sequences. Therefore, we limit our attention to the extension of the these methods to the dynamic image sequence. To this end, we propose a framework for detecting facial interest points based on the active shape model (ASM) [17] and then extracting the spatiotemporal features from the region components centered at these facial interest points for dynamic image sequences. Moreover, to reduce the feature dimensionality and select the more discriminative features, the AdaBoost method [18] is used for building robust learning models and for boosting our component-based approach.

The classifier design is another important issue in facial expression recognition. Most of the facial expression recognition approaches use only one classifier. Some studies [19][20] have shown that combining the output of several classifiers will lead to an improved classification performance, because each classifier makes errors on a different

region of the input space and multiple classifiers can supplement each other. According to our best knowledge, only few studies in facial expression recognition paid attention to multi-classifier fusion. To utilize the advantage of the multi-classifier fusion, in this paper, we also extend a framework of multi-classifier fusion based on decision rules to facial expression recognition.

In this paper, we propose a novel component-based approach for facial expression recognition from video sequences. Inspired by the methods presented in [15][21], 38 important facial interest regions based on prior information are first determined, and then spatiotemporal feature descriptors are used to describe facial expressions from these areas. Furthermore, we use AdaBoost to select the most important discriminative features for all components. In the classification step, we present a framework for fusing recognition results from several classifiers, such as support vector machines, boosting, Fisher discriminant classifier for exploiting the complementary information among different classifiers. Extensive experiments on the Cohn-Kanade facial expression database [22] are carried out to evaluate the performance of the proposed approach.

2 Boosted Component-Based Spatiotemporal Feature Descriptor

2.1 Facial Interest Points

In many earlier methods [1][2][23], fusion of geometric features and appearance features can improve the performance of expression recognizers. Geometric features are usually formed by parameters obtained by tracking facial action units or facial points' variation.

It is well known that not all features from the whole face are critical to expression recognizers. Yesin et al. [14] proposed to apply optical flow to regions based on positions of the eyes, eyebrows and the mouth. Zhang et al. [24] developed a framework in which Gabor wavelet coefficients were extracted from 34 fiducial points in the face image. In methods on scale-invariant feature transform (SIFT) [21], SIFT keypoints of objects are first extracted from a set of reference images in order to avoid from computing all points in an image. It is thus found that the search of interest points or regions in facial images is more important to component-based approach.

However, faces are different from other objects, in other words, important features for facial expression are always expressed in some special regions, such as mouth, cheek etc. Thus different from SIFT, our interest points detection is based on prior-experience. In our paper, 38 facial points are considered, shown in Fig. 1(a).

The approach for detecting those interest points is critical to our approach. If considering accuracy, manual labeling facial points for face image is good for expression recognizers. Unfortunately, this method costs much time and is not practical. It is well known that some methods are proposed for detecting or tracking facial points, such as AAM, ASM, and Elastic Bunch Graph Matching etc. After comparison, ASM [25] is applied to detect the facial points.

Geometric information from the first frame is obtained by applying ASM as shown in Fig. 1(a). Here, geometric models are trained from FRAV2D database [26] and MMI database [27].

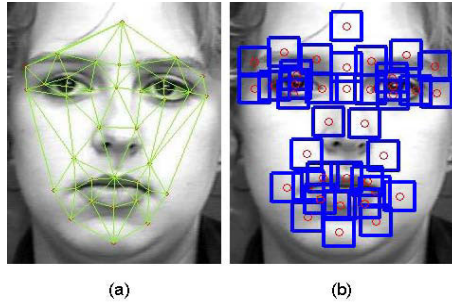


Fig. 1. (a) Results of facial points detection (b) Components for calculating spatiotemporal features

2.2 Component-Based Spatiotemporal Feature Descriptor

It is well known that feature extraction is critical to any facial expression recognition system. After detecting interest points, the appearance feature is considered next in our approach. Based on those facial interest points, the areas centered at these points have more discriminative information as shown in Fig. 1(b). The size of each area is 32×32 , it is observed that majority of features are focused on eyes and mouth. And the regions near cheeks and forehead are also considered in our approach. If the size of each area is too small, the features extracted from forehead, cheek, eyebrows have too little discriminative information. In contrast, if too large, most areas near mouth and eyes are overlapping too much, which would cause too much redundant information. In our experiments (Sec. 4), we will show the influence of region sizes.

LBP-TOP (local binary pattern from three orthogonal planes) has been proposed for motion analysis and shown excellent performance in the classification of expression recognition [28]. Features extracted by this method describe effectively appearance, horizontal motion and vertical motion from the image sequence.

We extend to use LBP-TOP to describe the spatiotemporal features of 38 components, shown in Fig. 2. In Fig. 2 XY plane shows the appearance of each component, XT plane shows the horizontal motion, which gives the idea of how one row changes in the temporal domain, YT as well shows the vertical motion, which gives the idea of how one column changes in the temporal domain. For LBP-TOP, it is possible to change the radii in axes X, Y and T, which are marked as R_X , R_Y and R_T . Also different numbers of neighboring points are used in the XY, XT and YT planes, which are marked as P_{XY} , P_{XT} and P_{YT} . Using these notions, LBP-TOP features are denoted as $\text{LBP-TOP}_{P_{XY}, P_{XT}, P_{YT}, R_X, R_Y, R_T}$. After detecting each component, the LBP-TOP histograms for each component are computed and concatenated into a single histogram to represent the appearance and motion of the facial expression sequence. In our further experiments, the radii in axes X, Y and T are set as 3; the numbers of local neighboring points around the central pixel for all three planes are set as 8. In our case, we use CSF (Component-based Spatialtemporal Features) for abbreviation.

The component detection of images with pose variation in a near-frontal view face is a challenge to our present implementation, since the component extraction is based on the first frame. For solving this problem, we use a simple solution to align face

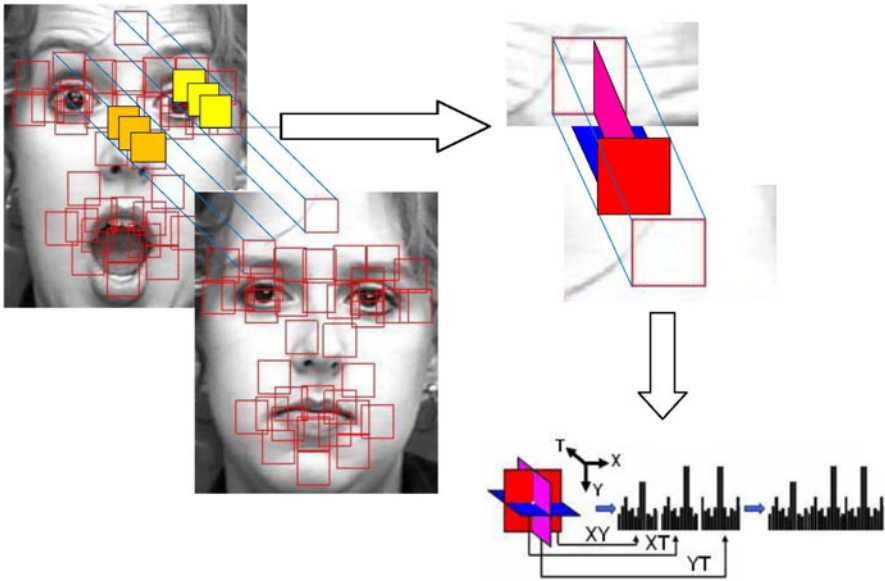


Fig. 2. Component-based spatiotemporal features

movement due to pose variation. For the first frame, ASM detects 38 facial interest points to provide 38 components for LBP-TOP. To reduce the effects of pose variations, ASM in the next frame detects the variation of eye coordinates. If the coordinates of the eyes are changed at a large extent compared with the former frame, the ASM is run again for providing new 38 facial components.

2.3 Boosted by AdaBoost

All components do not contain as much discriminative information as others for different expressions. It is not wise to use all information available in the image, but only the most important areas in terms of distinguishing between subjects or events. On the other hand, the dimensionality of the features extracted by CSF is quite high (38*59*3). Therefore, it is important to reduce the dimensionality of these features.

In order to select the different discriminative features for different expression pairs, we adopt AdaBoost and the concept of Intra-expression similarity and Extra-expression dissimilarity. In other words, the learners are designed for every expression-pair with an aim to learn more specific and discriminative features for each pair. Assume that the training samples P and Q belong to the i -th and j -th class, respectively. The dissimilarity of these samples is computed as

$$\chi_{P,Q} = \{\chi_{P,Q}^2(XY), \chi_{P,Q}^2(XT), \chi_{P,Q}^2(YT)\} \tag{1}$$

and the class for AdaBoost is labeled as +1 if $i = j$, otherwise labeled as -1. Next this dissimilarity and class information is fed into weak learners. Thus the AdaBoost algorithm selects features for discriminating the i -th and j -th class. In the same way, the features are learned for each expression-pair.

Here, the Chi square statistic was used as the dissimilarity measure of two LBP-TOP histograms computed from the components:

$$\chi^2_{P,Q} = \sum_{i=1}^L \frac{(P_i - Q_i)^2}{P_i + Q_i} \tag{2}$$

where P and Q are two CSF histograms, and L is the number of bins in the histogram.

3 Multi-classifier Fusion

We consider a C -class classification problem. A pattern described by CSF is, in general, a p -dimensional vector X . It is associated with a class label which can be represented by $\omega_t \in \{1, \dots, C\}$. Consider also the a posteriori probability function $P(\omega_t = i|X)$ represents the probability of the pattern X belonging to a given i -th class, given that X was observed. It is then natural to classify the pattern by choosing the j -th class with largest posteriori probability:

$$P(\omega_t = j|X) = \max_{i \in \{1, \dots, C\}} P(\omega_t = i|X) \tag{3}$$

Some studies [19,20] show better classification can be obtained if multiple classifiers are used instead of a single classifier. Consider that we have R classifiers (each representing the given pattern by a distinct measurement vector [19]), which are denoted as $D_k, k = 1, \dots, R$, for the same pattern X . In the k -th single classifier, its outputs are approximated by a posteriori probabilities $P(\omega_t = i|X)$, i.e.

$$P(\omega_t = i|D_k) = P(\omega_t = i|X) + \varepsilon_i(X) \tag{4}$$

where $\varepsilon_i(X)$ represents the error that a single classifier introduces.

From Eqn. 4, we consider a classifier that can approximate the a posteriori probability function $P(\omega_t = i|X)$, when $\varepsilon_i(X)$ is small. According to Bayesian theory, the pattern X should be assigned to the i -th class provided the a posteriori probability of that interpretation is maximum:

Assign $X \rightarrow \{\omega_t = j\}$ if

$$P(\omega_t = j|X, D_1, \dots, D_R) = \max_{i \in \{1, \dots, C\}} P(\omega_t = i|X, D_1, \dots, D_R) \tag{5}$$

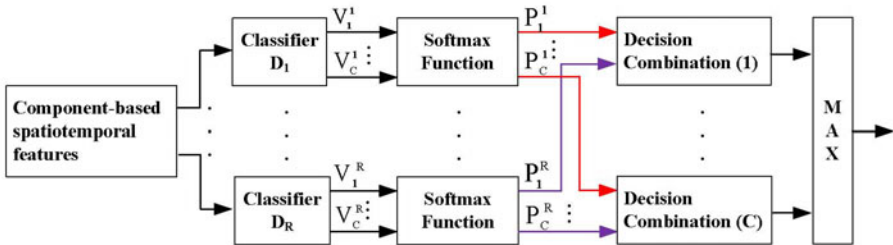


Fig. 3. The structure of multi-classifier fusion

For exploiting the complementary information among all classifiers, we investigated three decision rules (mean rule, product rule, and median rule). Detailed derivation of decision rules by Eqn. 5 and Bayesian theorem can be found e.g. in [19]. Assume that all classifiers used are generally statistically independent, and the priori probability of occurrence for i -th class model are under assumption of equal priors, the rule of multi-classifier fusion is simplified to

Assign $X \rightarrow \{\omega_t = j\}$ if

$$P(\omega_t = j|X, D_k) = \max_{i \in \{1, \dots, C\}} \text{DecisionRule} P(\omega_t = i|X, D_k) \quad (6)$$

As shown in Fig. 3 many popular classifiers, such as SVM, can output a voting vector which represents the voting numbers for each class. We denote V_i^k , for the voting number of i -th class from k -th classifier D_k .

These voting numbers are then converted to probabilities by applying the softmax function

$$P_i^k = P(\omega_t = i|X, D_k) = \frac{\exp(V_i^k)}{\sum_{i=1}^C \exp(V_i^k)} \quad (7)$$

Using this transformation does not change the classification decision for a classifier; moreover, it allows us to treat the classifier within Bayesian probabilistic framework.

4 Experiments

The proposed approach was evaluated with the Cohn-Kanade facial expression database. In our experiments, 374 sequences were selected from the database for basic expressions recognition. The sequences came from 97 subjects, with one to six expressions per subject.

Coordinates of facial fiducial points in the first frame are determined by ASM, and then the CSF features extracted from 38 facial components with fixed block size on those points are concatenated into one histogram. Ten-fold cross validation method was used in the whole scenario.

It was anticipated that the component size will influence the performance. Fig. 4 presents results using four block sizes with CSF. From this figure we can observe that the highest mean performance (94.92%) is reached when the component size is 16×16 , which was then selected for the following experiments.

AdaBoost is used to select the most important slices, as described in Sec. 2.3. In our experiments, the number of slices varies at 15, 30, 45, 60, 75, 90. The average recognition accuracies corresponding to different number of slices are 90.37%, 91.98%, 94.12%, 93.32%, 93.05%, 92.25%, respectively. It is observed that the best accuracy of 94.12% is obtained with 45 slices. Compared with the result in Fig. 4 at optimal block size, the accuracy decreases by 0.8%, but the dimensionality of the feature space is reduced from $38 \times 59 \times 3$ (6726) to 45×59 (2655).

The six-expression classification problem was decomposed into 15 two-class problems. Therefore, each test sample is classified by 15 expression-pair sub-classifiers. In multi-classifier fusion, 15 sub-classifiers as a whole were thought as an individual classifier D_k as shown in Fig. 3. After selecting the optimal component size, five different

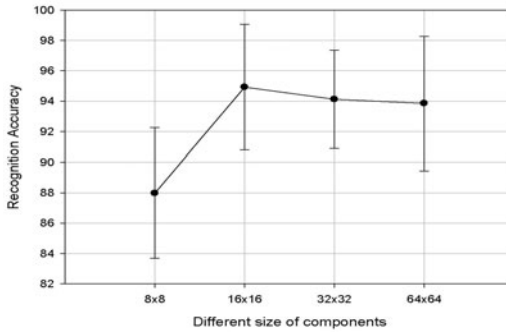


Fig. 4. Performance comparison (%) with features from different size of components

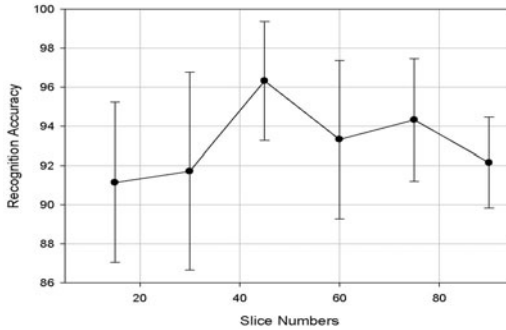


Fig. 5. Performance comparison (%) using AdaBoost on different slice numbers with multi-classifier fusion (using median rule)

classifiers, i.e. three support vector machines (SVM) based on linear kernel, gaussian kernel and poly kernel, a boosting classifier (Boosting) and a Fisher linear discriminant classifier (FLD) were chosen as individual classifiers. These classifiers performed better in our experiments than a Bayesian classifier and k-nearest neighbor classifier.

For boosting the performance of each individual classifier, three decision rules, i.e. median rule, mean rule, and product rule, are investigated for multi-classifier fusion. The average accuracies are 94.39%, 95.19%, and 94.39% for the mean, median and product rule, respectively. Comparing with Fig. 4, the performance of multi-classifier fusion is increased by 0.27% when using the median rule, while the two other rules cannot boost the performance of individual classifiers for this dataset.

Fig. 5 lists the results for feature selection by AdaBoost on different number of slices with multi-classifier fusion (using median rule). It can be observed that the average performance gets the best rate (96.32%) with 45 slices.

Table 1 compares our methods: CSF, CSF with multi-classifier fusion (CSFMC), Boosted CSF with multi-classifier fusion (BCSFMC), and some other methods, providing the overall results obtained with Cohn-Kanade Database in terms of the number of people (PN), the number of sequences (SN), expression classes (CN), with different

Table 1. Comparison with different approaches

Method	PN	SN	SN	Decision Rule	Measure	Recognition Rate(%)
[14]	97	-	6	-	Five-fold	90.9
[29]	96	320	7(6)	-	Ten-fold	88.4
[30]	90	284	6	-	-	93.66
[31]	90	313	7	-	Leave-One-Subject-Out	93.8
[32]	97	374	6	-	Ten-fold	91.44
CSF	97	374	6	-	Ten-fold	94.92
CSFMC	97	374	6	Median rule	Ten-fold	95.19
BCSFMC	97	374	6	Median rule	Ten-fold	96.32

measures. It should be noted that the results are not directly comparable due to different experimental setups, processing methods, the number of sequences used etc., but they still give an indication of the discriminative power of each approach. From this table, we can see that CSF obtained better result than block-based LBP-TOP that divided face image into 8×8 overlapping blocks [32], with an increase of 3.48%. Additionally, CSFMC and BCSFMC are slightly better compared to CSF. BCSFMC outperformed all the other methods.

5 Conclusion

In order to boost facial expression recognition, we propose a component-based spatiotemporal feature (CSF) to describe facial expressions from video sequences. In our approach, facial interest points in an initial frame are detected by ASM that is robust to errors in fiducial point localization. According to those interest points, facial components are computed on areas centered at those points in a sequence, providing less redundant information than block-based methods. Comparing with appearance and geometric approaches, our component-based spatiotemporal approach belongs to hybrid methods with advantages from both. However, our method describes the dynamic features from video sequences. Furthermore, for boosting CSF and reducing the computational cost of each classifier, AdaBoost is utilized to select the most discriminative spatiotemporal slices from the facial components. Finally, we also present an approach for fusing several individual classifiers based on mean, median or product rule.

In experiments on the Cohn-Kanade database we have demonstrated that the CSF descriptors with multi-classifier fusion and AdaBoost feature selection lead to a promising improvement in facial expression classification. In future work we plan to explore how our approach could be adopted to very challenging problems including more severe head pose variations and occlusion. Spontaneous facial expressions common in many practical applications of facial expression recognition will also be studied.

Acknowledgements

The financial support provided by the Academy of Finland is gratefully acknowledged. The first author is funded by China Scholarship Council of Chinese government. This

work was partly supported by National Natural Science Foundation of China under Grants 60872160 and 61073137. The authors would like to thank the anonymous reviewers for their constructive advice.

References

1. Fasel, B., Luetttin, J.: Automatic facial expression analysis: a survey. *Pattern Recognition* 36, 259–275 (2003)
2. Zeng, Z., Pantic, M., Roisman, G., Huang, T.: A survey of affective recognition methods: Audio, visual and spontaneous expression. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31(1), 39–58 (2009)
3. Tian, T., Kanade, T., Cohn, J.: Facial expression analysis. In: Li, S., Jain, A.K. (eds.) *Handbook of Face Recognition*. Springer, Heidelberg (2004)
4. Tian, Y., Kanade, T., Cohn, J.: Recognizing action units for facial expression analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23(2), 97–115 (1999)
5. Kobayashi, H., Hara, F.: Facial interaction between animated 3D face robot and human being. In: *Systems, Man and Cybernetics*, pp. 3732–3737. IEEE Press, New York (1997)
6. Lyons, M., Akamatsu, S., Kamachi, M., Gyoba, J.: Coding facial expressions with Gabor wavelets. In: *Automatic Face and Gesture Recognition*, pp. 200–205. IEEE Press, New York (1998)
7. Yaser, Y., Larry, S.: Recognizing human facial expressions from long image sequences using optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18(6), 636–642 (1996)
8. Chen, F., Kotani, K.: Facial expression recognition by supervised independent component analysis using MAP estimation. *IEICE - Transactions on Information and Systems* 2, 341–350 (2008)
9. Penev, P., Atick, J.: Local feature analysis: a general statistical theory for object representation. *Network: Computation in Neural Systems* 7(3), 477–500 (1996)
10. Kotsia, I., Pitas, I.: Facial expression recognition in image sequences using geometric deformation features and support vector machines. *IEEE Transactions on Image Processing* 16(1), 172–187 (2007)
11. Donato, G., Bartlett, M., Hager, J., Ekman, P., Sejnowski, T.: Classifying facial actions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21(10), 974–989 (1999)
12. Feng, X., Pietikäinen, M., Hadid, A.: Facial expression recognition based on local binary patterns and linear programming. *Pattern Recognition and Image Analysis* 15(2), 546–548 (2005)
13. Lanitis, A., Taylor, C., Cootes, T.: Automatic interpretation and coding of face images using flexible models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19(7), 743–756 (1997)
14. Yesin, M., Bullot, B., Sharma, R.: From facial expression to level of interest: a spatio-temporal approach. In: *Computer Vision and Pattern Recognition*, pp. 922–927. IEEE Press, New York (2004)
15. Heisele, B., Koshizen, B.: Components for face recognition. In: *Automatic Face and Gesture Recognition*, pp. 153–158. IEEE Press, New York (2004)
16. Ivanov, I., Heisele, B., Serre, T.: Using component features for face recognition. In: *Automatic Face and Gesture Recognition*, pp. 421–426. IEEE Press, New York (2004)
17. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.G.: Active shape models - their training and application. *Computer Vision and Image Understanding* 61, 38–59 (1995)

18. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to Boosting. In: Computational Learning Theory: Eurocolt 1995, pp. 23–37 (1995)
19. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On combining classifiers. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 20(3), 226–239 (1998)
20. David, M.J., Martijin, B.V., Robert, P.W.D., Josef, K.: Combining multiple classifiers by averaging or by multiplying? *Pattern Recognition* 33, 1475–1485 (2000)
21. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60(2), 91–110 (2004)
22. Kanade, T., Cohn, J.F., Tian, Y.: Comprehensive database for facial expression analysis. In: Automatic Face Recognition and Gesture Recognition, pp. 46–53. IEEE Press, New York (2000)
23. Pantic, M., Patras, I.: Dynamic of facial expression: Recognition of facial actions and their temporal segments from face profile image sequences. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*. 36(2), 433–449 (2006)
24. Zhang, Z., Lyons, M., Schuster, M., Akamatsu, S.: Comparison between geometry-based and Gabor-wavelets-based facial expression recognition using multi-layer perceptron. In: International Workshop on Automatic Face and Gesture Recognition, pp. 454–459. IEEE Press, New York (1998)
25. Milborrow, S., Nicolls, F.: Locating facial features with extended active shape model. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 504–513. Springer, Heidelberg (2008)
26. Serrano, A., Diego, I.M., Conde, C., Cabello, E.: Influence of wavelet frequency and orientation in an SVM-based parallel Gabor PCA face verification system. In: Yin, H., Tino, P., Corchado, E., Byrne, W., Yao, X. (eds.) IDEAL 2007. LNCS, vol. 4881, pp. 219–228. Springer, Heidelberg (2007)
27. MMI Database, <http://www.mmifacedb.com>
28. Zhao, G., Pietikäinen, M.: Dynamic texture recognition using local binary pattern with an application to facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29(6), 915–928 (2007)
29. Shan, C., Gong, S., McOwan, P.W.: Robust facial expression recognition using local binary patterns. In: Image Processing, pp. 370–373. IEEE Press, New York (2005)
30. Aleksic, S., Katsaggelos, K.: Automatic facial expression recognition using facial animation parameters and multi-stream HMMS. *IEEE Transactions on Information Forensics and Security* 1(1), 3–11 (2006)
31. Littlewort, G., Bartlett, M., Fasel, I., Susskind, J., Movellan, J.: Dynamics of facial expression extracted automatically from video. In: IEEE Workshop Face Processing in Video. IEEE Press, New York (2004)
32. Zhao, G., Pietikäinen, M.: Boosted multi-resolution spatio temporal descriptors for facial expression recognition. *Pattern Recognition Letters* 30, 1117–1127 (2009)

Gender Classification on Real-Life Faces

Caifeng Shan*

Philips Research
High-Tech Campus 36, 5656AE Eindhoven, The Netherlands
caifeng.shan@philips.com

Abstract. Gender recognition is one of fundamental tasks of face image analysis. Most of the existing studies have focused on face images acquired under controlled conditions. However, real-world applications require gender classification on real-life faces, which is much more challenging due to significant appearance variations in unconstrained scenarios. In this paper, we investigate gender recognition on real-life faces using the recently built database, the Labeled Faces in the Wild (LFW). Local Binary Patterns (LBP) is employed to describe faces, and Adaboost is used to select the discriminative LBP features. We obtain the performance of 94.44% by applying Support Vector Machine (SVM) with the boosted LBP features. The public database used in this study makes future benchmark and evaluation possible.

Keywords: Gender Classification, Local Binary Patterns, AdaBoost, Support Vector Machines.

1 Introduction

Gender classification is a fundamental task for human beings, as many social functions critically depend on the correct gender perception. Automatic gender recognition has many potential applications, for example, shopping statistics for marketing, intelligent user interface, visual surveillance, etc. Human faces provide important visual information for gender perception. Gender classification from face images has received much research interest in last two decades.

In the early 1990s various neural network techniques were employed to recognize gender from frontal faces [1,2], for example, Golomb *et al.* [1] trained a fully connected two-layer neural network, SEXNET, which achieves the recognition accuracy of 91.9% on 90 face images. Recent years have witnessed many advances (e.g., [3,4]); we summarize recent studies in Table 1. Moghaddam and Yang [5] used raw image pixels with nonlinear Support Vector Machines (SVMs) for gender classification on thumbnail faces (12×21 pixels); their experiments on the FERET database (1,755 faces) demonstrated SVMs are superior to other classifiers, achieving the accuracy of 96.6%. In [6], local region matching and holistic features were exploited with Linear Discriminant Analysis (LDA) and SVM for gender recognition. On the 12,964 frontal faces from multiple databases

* Supported by the Visual Context Modelling (ViCoMo) project.

Table 1. Overview of recent studies on gender classification from face images

Study	Data Set			Approach		Performance
	Data	Real-Life	Public	Feature	Classifier	
2002 [5]	1,755	No	Yes	raw pixels	SVM	96.62%
2002 [10]	3,500	Yes	No	haar-like features	Adaboost	79.0%
2005 [6]	12,964	No	Yes	local-region matching	SVM	94.2%
2006 [7]	5,326	No	Yes	fragment-based filter banks	boosting	91.72%
2007 [8]	2,409	No	Yes	pixel comparisons	Adaboost	94.3%
2008 [9]	500	No	Yes	raw pixels	SVM	86.54%
2009 [11]	10,100	Yes	No	haar-like features	probabilistic boosting tree	95.51%
our work	7,443	Yes	Yes	boosted LBP features	SVM	94.44%

(including FERET and PIE), local region-based SVM achieved the performance of 94.2%. Lapedriza *et al.* [7] compared facial features from internal zone (eyes, nose, and mouth) and external zone (hair, chin, and ears). Their experiments on the FRGC database show that the external face zone contributes useful information for gender classification. Baluja and Rowley [8] introduced an efficient gender recognition system by boosting pixel comparisons in face images. On the FERET database, their approach matches SVM with 500 comparison operations on 20×20 pixel images. Mäkinen and Raisamo [9] systematically evaluated different face alignment and gender recognition methods on the FERET database.



Fig. 1. Examples of real-life faces (from the LFW database). (*top 2 rows*) Female; (*bottom 2 rows*) Male.

A common problem of the above studies is that face images acquired under controlled conditions (e.g., FERET database) are considered, which usually are frontal, occlusion-free, with clean background, consistent lighting, and limited facial expressions. However, in real-world applications, gender classification needs to be performed on real-life face images captured in unconstrained scenarios; see Fig. 1 for examples of real-life faces. As can be observed, there are significant appearance variations in real-life faces, which include facial expressions, illumination changes, head pose variations, occlusion or make-up, poor image quality, and so on. Therefore, gender recognition on real-life faces is much more challenging compared to the case of faces captured in constrained environments. Few studies in the literature have addressed this problem. Shakhnarovich *et al.* [10] made an early attempt by collecting over 3,500 face images from the web. On this difficult data set, using Haar-like features, they obtained the performance of 79.0% (Adaboost) and 75.5% (SVM). More recently Gao and Ai [11] adopted the probabilistic boosting tree with Haar-like features, and obtained the accuracy of 95.51% on 10,100 real-life faces. However, the data sets used in these studies are not publicly available; therefore, it is difficult to use them as benchmark in research.

In this paper, we focus on gender recognition on real-life faces. Specifically, we use a recently built public database, the Labeled Faces in the Wild (LFW) [12]. To the best of our knowledge, this is the first study about gender classification on this difficult database. Local Binary Patterns (LBP) [13] is employed to extract facial features. We adopt Adaboost to learn the most discriminative LBP features, which, when used with SVM, provide the performance of 94.44%. The public database used in this study enables future benchmark and evaluation.

2 Gender Recognition

2.1 Data Set

The Labeled Faces in the Wild is a database for studying the problem of unconstrained face recognition, which contains 13,233 color face photographs of 5,749 subjects collected from the web. Fig. 1 shows example images in the database. All the faces were detected by the Viola-Jones face detector, and the images are centered using detected faces and scaled to the size of 250×250 pixels.

We manually labeled the ground truth regarding gender for each face. We did not consider the faces that are not (near) frontal, as well as those for which it is difficult to establish the ground truth. Some examples of the removed faces are shown in Fig. 2. In our experiments, we chose 7,443 face images (2,943 females and 4,500 males); see Fig. 1 for some examples. As illustrated in Fig. 3, all images were aligned with a commercial face alignment software [14], and then the grayscale faces of 127×91 pixels were cropped from aligned images for use. The data set we used will be shared online for public benchmark and evaluation.

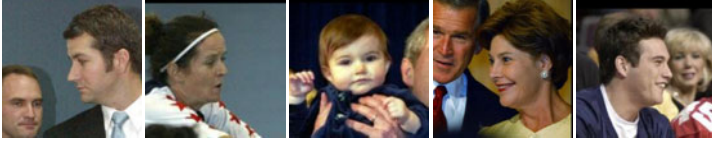


Fig. 2. Example images that are not considered



Fig. 3. The pre-processing process on face images. (*left*) original image; (*middle*) aligned image; (*right*) cropped face.

2.2 Our Approach

A gender recognition system consists of two key components: facial feature extraction and classifier design. As reviewed in Section 1 and Table 1, raw image pixels and Haar-like features are two often-used representations. In this work, we employ LBP features, which have been widely exploited for facial representation in recent years [13]. The most important properties of LBP features are their tolerance against monotonic illumination changes and their computational simplicity. The LBP operator labels the image pixels by thresholding a neighborhood of each pixel with the center value and considering the results as a binary number. As shown in Fig. 4, face images are divided into non-overlapping sub-regions, and the LBP histograms extracted from each sub-region are concatenated into a feature histogram. Following the parameter settings suggested in [13], in our experiments, face images of 127×91 pixels were divided into 42 sub-regions of 18×15 pixels, and the 59-label $LBP(8, 2, u_2)$ operator [13] was adopted to extract LBP features. Thus each face image was described by a LBP histogram of 2,478 (42×59) bins. With the LBP-based representation, SVM can be used for gender classification, which has been an effective classification method in existing studies [5,9].

Beyond the above standard representation, we further adopt Adaboost to learn the discriminative LBP-Histogram (LBPH) bins for gender classification. Adaboost has proved effective in both accuracy and speed for gender classification [10,8]. Here we aim to select the LBPH bins which best separate the female and male samples. The weak classifier $h_j(x)$ consists of a feature f_j which corresponds to the LBPH bin, a threshold θ_j and a parity p_j indicating the direction of the inequality sign:

$$h_j(x) = \begin{cases} 1 & \text{if } p_j f_j(x) \leq p_j \theta_j \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

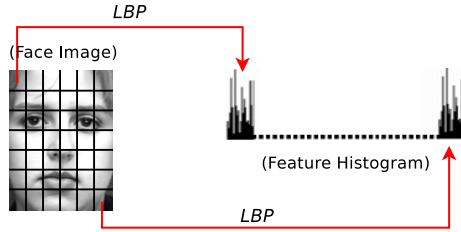


Fig. 4. Each face image is divided into sub-regions from which LBP histograms are extracted and concatenated into a single, spatially enhanced feature histogram

In [15], Adaboost was used to select the discriminative sub-regions (in terms of LBP histogram) from a large pool generated by shifting and scaling a sub-window over face images. In contrast, here we look at regional LBP histograms at the bin level, to identify the discriminative LBPH bins.

3 Experiments

All experimental results were obtained using the 5-fold cross-validation. We partitioned the data set into five subsets of similar size, with a similar balance between the two classes. The images of a particular subject appear only in one subset. In each trial, one subset was used for testing, while the remaining four subsets were used for training. The recognition results were averaged over the 5 trials.

In our experiments, we used the SVM implementation in the library SPIDER¹. The Radius Basis Function (RBF) kernel was utilized, and the parameters were tuned to obtain the best performance. Meanwhile, each dimension of the feature vector was scaled to be between -1 and 1. As a baseline to compare against, we also applied SVM with raw image pixels, which delivers the best performance on face images acquired in controlled environments [5]. For computational simplicity, face images of 127×91 pixels were down-scaled to 64×46 pixels, thus each image represented by a vector of 2,944 dimensions. We summarize the results of SVM with raw pixels and standard LBP features in Table 2. As can be observed, LBP features produce better performance than raw image pixels. Regarding support vectors, with raw pixels, the learned SVMs utilized 51-53% of the total number of training samples (in each trial of cross-validation, the number varies slightly), while SVMs with LBP features employ 58-61%.

For boosting learning, to generate a large LBP feature pool, we can generate many more sub-regions by shifting and scaling a sub-window over face images. In this study, we fixed the size of sub-window as 18×15 pixels, and shifted the sub-window with the shifting step of 4 pixels vertically and 3 pixels horizontally. In total 700 sub-regions were obtained. By applying 59-label $LBP(8, 2, u_2)$ to each sub-regions, a histogram of 413,000 (700×59) bins was extracted from each face image. We adopted Adaboost to learn discriminative LBPH bins and boost

¹ <http://www.kyb.tuebingen.mpg.de/bs/people/spider/index.html>

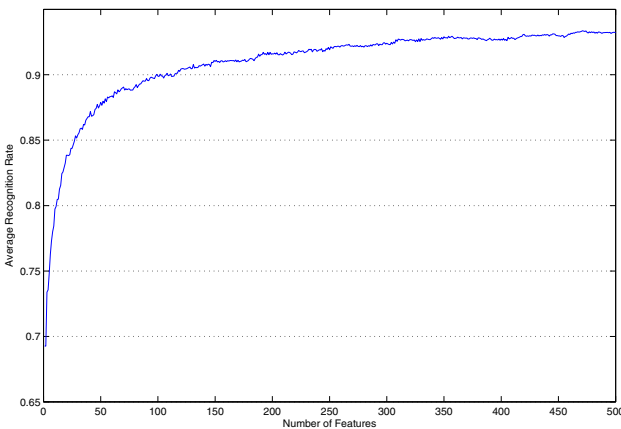
Table 2. Experimental results of gender classification

Approach			Recognition Rates (%)		
Feature	Dim.	Classifier	Female	Male	Overall
raw pixels	2,944	SVM	86.89	94.13	91.27±1.67
standard LBP	2,478	SVM	89.78	95.73	93.38±1.50
boosted LBP	500	Adaboost	91.13	94.82	93.36±1.49
boosted LBP	500	SVM	91.91	96.09	94.44±1.19

a strong classifier. We plot in Fig. 5 the average accuracy of Adaboost as a function of the number of features selected. With the 500 selected LBPH bins, Adaboost achieves recognition rate of 93.36%, which is comparable to that of SVM using the standard LBP (2,478 bins). However, Adaboost is much more computationally efficient than SVM, requiring much less features.

We plot in the left side of Fig. 6 the top 20 sub-regions that contain most LBPH bins selected. The right side of Fig. 6 further shows the spatial distribution of the selected 500 LBPH bins in the 5-fold cross-validation experiments, where each small patch represents the corresponding sub-region, and the grayscale intensity of each patch is proportional to the number of bins selected from that sub-region. It is observed that the discriminative LBPH bins are mainly distributed in the regions around/above eyes. Although faces are (on average) symmetric, the selected features are not symmetric, because of the pose/illumination variations in the dataset. Regarding the distribution of selected features among 59 bins, we plot in Fig. 7 the distribution of the 500 features selected. We can observe that selected bins distribute in all 59 bins, but some bins do have more contributions (e.g., bin 2, 12, 27, and 34).

We further adopted SVM with the selected LBPH bins for gender classification, which achieves the best performance of 94.44%. Moreover, the numbers of support vectors were 32-35% of the total number of training samples, which are

**Fig. 5.** Average recognition rate of Adaboost, as a function of the number of feature used

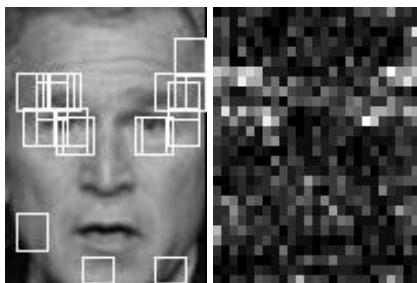


Fig. 6. (*left*) The top 20 sub-regions that contain most LBPH bins selected; (*right*) the spatial distribution of the 500 LBPH bins selected

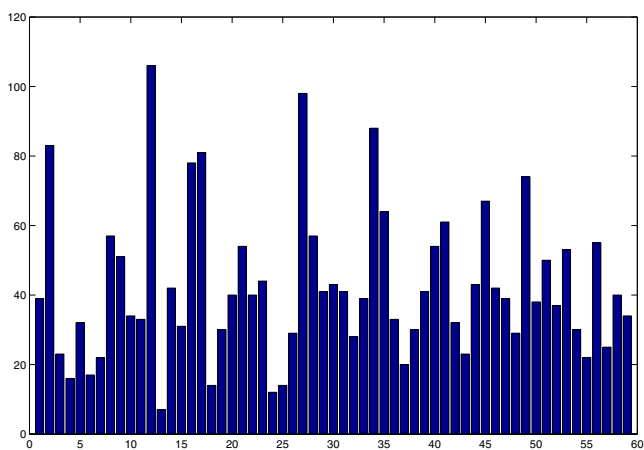


Fig. 7. The distribution of the selected 500 LBP features



Fig. 8. Examples of failure on gender recognition. (*top*): female mis-recognized as male; (*bottom*): male mis-recognized as female.

much less than those of SVMs using raw pixels or standard LBP. As observed in Table 2, the boosted LBP based SVM also produces the smallest standard variation, thus more robust than other methods. We see in Table 2 there is notable bias towards males in all experiments. This is also observed in existing studies [10]. Finally we show in Fig. 8 some examples of mis-classification, some of which could be due to pose variations, occlusion (e.g., glasses), and facial expressions.

4 Conclusions

In this paper, we investigate gender recognition from faces acquired in unconstrained conditions. Extensive experiments have been conducted on the LFW database. We adopted Adaboost to learn the discriminative LBP features, and SVM with boosted LBP features achieves the accuracy of 94.44% on this difficult database.

References

1. Golomb, B.A., Lawrence, D.T., Sejnowski, T.J.: Sexnet: A neural network identifies sex from human faces. In: *Advances in Neural Information Processing Systems, NIPS (1991)*
2. Brunelli, R., Poggio, T.: Hyperbf networks for gender classification. In: *DRAPA Image Understanding Workshop (1992)*
3. Yang, Z., Li, M., Ai, H.: An experimental study on automatic face gender classification. In: *International Conference on Pattern Recognition (ICPR)*, pp. 1099–1102 (2006)
4. Hadid, A., Pietikäinen, M.: Combining appearance and motion for face and gender recognition from videos. *Pattern Recognition* 42(11), 2818–2827 (2009)
5. Moghaddam, B., Yang, M.: Learning gender with support faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(5), 707–711 (2002)
6. BenAbdelkader, C., Griffin, P.: A local region-based approach to gender classification from face images. In: *Computer Vision and Pattern Recognition Workshop*, pp. 52–52 (2005)
7. Lapedriza, A., Marin-Jimenez, M.J., Vitria, J.: Gender recognition in non controlled environments. In: *International Conference on Pattern Recognition (ICPR)*, pp. 834–837 (2006)
8. Baluja, S., Rowley, H.A.: Boosting set identification performance. *International Journal of Computer Vision (IJCV)* 71(1), 111–119 (2007)
9. Mäkinen, E., Raisamo, R.: Evaluation of gender classification methods with automatically detected and aligned faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30(3), 541–547 (2008)
10. Shakhnarovich, G., Viola, P.A., Moghaddam, B.: A unified learning framework for real time face detection and classification. In: *IEEE International Conference on Automatic Face & Gesture Recognition (FG 2002)*, pp. 14–21 (2002)
11. Gao, W., Ai, H.: Face gender classification on consumer images in a multiethnic environment. In: *International Conference on Biometrics (ICB)*, pp. 169–178 (2009)
12. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: A database for studying face recognition in unconstrained environments. *Tech. Rep. 07-49*, University of Massachusetts, Amherst (October 2007)

13. Ahonen, T., Hadid, A., Pietikäinen, M.: Face recognition with local binary patterns. In: Pajdla, T., Matas, J.(G.) (eds.) *ECCV 2004*. LNCS, vol. 3021, pp. 469–481. Springer, Heidelberg (2004)
14. Wolf, L., Hassner, T., Taigman, Y.: Similarity scores based on background samples. In: Zha, H., Taniguchi, R.-i., Maybank, S. (eds.) *Computer Vision – ACCV 2009*. LNCS, vol. 5996. Springer, Heidelberg (2010)
15. Sun, N., Zheng, W., Sun, C., Zou, C., Zhao, L.: Gender classification based on boosting local binary pattern. In: *International Symposium on Neural Networks*, pp. 194–201 (2006)

Face Recognition Using Contourlet Transform and Multidirectional Illumination from a Computer Screen

Ajmal Mian

School of Computer Science and Software Engineering
The University of Western Australia
35 Stirling Highway, Crawley, WA 6009, Australia
ajmal@csse.uwa.edu.au

Abstract. Images of a face under arbitrary distant point light source illuminations can be used to construct its illumination cone or a linear subspace that represents the set of facial images under all possible illuminations. However, such images are difficult to acquire in everyday life due to limitations of space and light intensity. This paper presents an algorithm for face recognition using multidirectional illumination generated by close and extended light sources, such as the computer screen. The Contourlet coefficients of training faces at multiple scales and orientations are calculated and projected separately to PCA subspaces and stacked to form feature vectors. These vectors are projected once again to a linear subspace and used for classification. During testing, similar features are calculated for a query face and matched with the training data to find its identity. Experiments were performed using in house data comprising 4347 images of 106 subjects and promising results were achieved. The proposed algorithm was also tested on the extended Yale B and CMU-PIE databases for comparison of results to existing techniques.

1 Introduction

Face recognition under varying illumination is a challenging problem because the appearance of a face changes dramatically with illumination. In fact, changes due to illumination can be greater than the changes due to face identity. Other variations due to pose and facial expressions can introduce further challenges however, they are less problematic when the subject is cooperative. In this paper, we consider variations due to illumination alone.

Face recognition is extensively studied due to its potential applications in security, surveillance and human computer interaction. Zhao et. al. [26] provide a detailed survey of face recognition literature and categorize them into holistic face recognition techniques which match global features of the complete face [23, 3], feature-based techniques which match local features of the face [25] and hybrid techniques which use both holistic and local features. From the perspective of data, face recognition can be divided into appearance based or shape based techniques. While appearance based techniques are considered sensitive

to illumination variations, there are claims that 3D face recognition is illumination invariant. Although 3D faces are illumination invariant once the data has been acquired, the data acquisition process itself is not illumination invariant. This is because accurate 3D face data requires active illumination from a laser or a projector. Moreover, changes in ambient illumination can still have a great impact on the accuracy and completeness of 3D data. Dark regions such as eyebrows and specularities can cause missing data or spikes. These problems are discussed in detail by Bowyer et al. [5] in their survey of 3D face recognition.

In search of illumination invariance, Chu et al. [7] proposed active frontal illumination from NIR LEDs for face recognition. This approach has the advantage of being invariant to ambient lighting and the NIR illumination is imperceptible to the eye. However, like 3D face recognition, this approach is not truly illumination invariant as it relies on active illumination and custom hardware.

The human visual perception has inspired many researchers to use video or image sequences to construct a joint representation of the face in spatial and temporal space for identification [26]. A single image contains spatial information but the temporal dimension defines trajectories of facial features and body motion characteristics which may further assist classification. Arandjelovic and Cipolla [1] proposed shape-illumination manifolds to represent a face under changing illumination conditions. They first find the best match to a video sequence in terms of pose and then re-illuminate them based on the manifold. Appearance manifolds under changing pose were also used by Lee and Kriegman [12] to perform face recognition. Both approaches assume the presence of pose variations which imply image acquisition over longer durations.

Li et al. [14] extracted the shape and pose free facial texture patterns from multi-view face images and used KDA for classification. Liu et al. [16] perform online learning for multiple image based face recognition without using a pre-trained model. Tangelder and Schouten [22] used a sparse representation of multiple still images for face recognition. A common aspect of existing multiple image/video-based techniques is that they rely on changes in pose or long term changes to extract additional information which implies longer acquisition times. An underlying assumption is that the images must contain non-redundant information either due to the relative motion of the camera and the face or the motion of the facial features due to expressions. Multiple images of a face acquired instantly e.g. 10 frames/sec, from a fixed viewpoint, will be mostly redundant and the temporal dimension will not contain any additional information.

It is possible to instantly acquire non-redundant images by changing the illumination. Belhumeur and Kriegman used multiple images under arbitrary point source illuminations to construct the 3D shape of objects [4]. Lee et al. [9] extended the idea to construct 3D faces and its corresponding albedo and subsequently used them to synthesize a large number (80-120) of facial images under novel illuminations. The synthetic images were used to estimate the illumination cone of the face for illumination invariant face recognition. Hallinan [10] empirically showed that the illumination cone can be approximated by a five dimensional subspace. Basri and Jacobs [2] showed that the illumination cone of

convex Lambertian surfaces can be approximated by a nine dimensional linear subspace. According to Lee et al.'s interpretation [13], there exist nine universal *virtual* lighting conditions such that the images under these illuminations are sufficient to approximate its illumination cone. Lee et al. [13] showed that a linear subspace can be constructed from nine physical lighting conditions that provides a good representation for illumination invariant face recognition. With nine physical lighting directions, the need for 3D face construction and albedo required by [9] [2] can be avoided. However, some of the light source directions suggested in [13] are at angles greater than 100 degrees. Distant light sources at such angles are difficult to achieve in practical situations due to space limitations.

Another difficulty with point light sources is that they must be of significantly high intensity. Schechner et al. [18] showed that images under multiplexed illumination of a collection of point light sources can solve this problem by offering better signal to noise ratio. The results of Lee et al. [13] suggest that the superposition of images under different point source lighting or images with a strong ambient component are more effective for face recognition. These findings naturally hint towards studying face recognition under extended light sources which is the focus of our research. In this paper, we try to answer the question: Is it possible to construct a subspace representation of the face for illumination invariant face recognition using extended light sources? Besides, minimizing the need for space, the proposed face recognition algorithm is designed with the following practical constraints. (1) Use of desktop/office equipment and no custom hardware. (2) Minimization of the number of training images. (3) Minimization of representation/memory requirements.

Unlike distant point light source, extended light source implies that it will not essentially form a constant vector towards all points on the face. Thus standard photometric stereo techniques cannot be used in this case and neither can the illumination cone be estimated. However, on the bright side, extended light sources can be placed close to the face alleviating the need for large space and high brightness. In our setup, illumination is varied by scanning a horizontal and then a vertical white stripe (with black background) on the computer screen in front of the subject. Fig. 1 shows an illustration of our approach. The Contourlet coefficients [8] of the images at different scales and orientations are projected separately to PCA subspaces and then stacked to form a feature vector. These features are projected once again to a linear subspace and used for classification.

Our setup was initially proposed in [17] where we used 47 images. In this paper, we drop the number of images to 23 because adjacent images had quite similar illumination in [17]. In [17], we constructed two global space-time representations using multiple images per face and sliding windows to match the two representations to the database separately. In this paper, a single image per face is used to construct its spatial representation and multiple representations are used to construct a subspace for training a classifier. Hence, recognition is performed using a single image. The database has been increased from 10 to 106 subjects and comparison with other techniques is performed on the extended Yale B and CMU-PIE databases.

2 Subspace Feature Representation

Fig. 1 shows our image acquisition setup. For a good signal to noise ratio, the subject must not be far from the screen. The camera's output is displayed on the screen so that the subject can approximately center his/her face. Image capture is automatically initiated [24] when the face is correctly positioned, or it can be manually initiated. A white horizontal stripe scans from the top to bottom of the screen followed by a white vertical stripe which scans from left to right. In our experiments, the stripe was 200 pixels thick and 8 images were captured during vertical scan and 15 during horizontal scan (given the aspect ratio of the screen). A final image was captured in ambient light for subtracting from all other images if required. All images are normalized so that a straight horizontal line passes through the center of their eyes. The scale of the images is also normalized based on the manually identified centers of eyes and lips. This normalization is similar to the normalization used for Yale B database in [9]. The manual identification of eyes and lips can be replaced with automatic eyes and lips detection which can be accurately performed on the basis of all 23 images given that they are captured instantly without subject movement. See Fig. 2 for sample images. A mask was used to remove the lower corners of the image. We imaged 106 subjects over a period of eight months. Out of these, 83 were imaged in two different sessions with an average of 60 days gap.

We construct the subspaces in the feature space and use the Contourlet transform [8] for extracting features. The Contourlet transform is an extension of Wavelets. Gabor wavelets have been well studied for face recognition and many variants exist [25] [27] [15]. A survey of wavelets based face recognition is given in



Fig. 1. Multiple images of a subject are acquired while illumination is varied by moving a white stripe on a computer screen

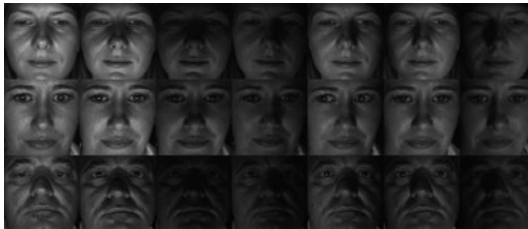


Fig. 2. Sample faces after preprocessing

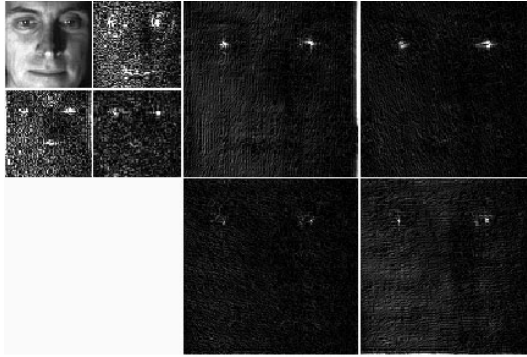


Fig. 3. Contourlet coefficients of a sample face

[19]. Wavelets provide a time-frequency representation of signals and are good at analyzing point (or zero dimensional) discontinuities. Therefore, Wavelets are suitable for analyzing one dimensional signals. On the other hand, images are inherently two dimensional and can have one dimensional discontinuities such as curves. These discontinuities can be captured by Contourlets [8]. The Contourlet transform performs multi-resolution and multi-directional decomposition of images allowing for different number of directions at each scale [8].

Let \mathbf{a}_i^{sk} represent the vector of Contourlet coefficients of the i th image (where $i = 1 \dots 23$) at scale s and orientation k . The Contourlet transform has 33% inherent redundancy [8]. Moreover, the Contourlet coefficients (at the same scale and orientation) of many faces can be approximated by a much smaller linear subspace. Therefore, the Contourlet coefficients of all training images calculated at the same scale and orientation are projected separately to PCA subspaces.

Let $\mathbf{A}^{sk} = [\mathbf{a}_{ij}^{sk}]$ (where $i \in \{1, 2 \dots 23\}$, and $j = 1, 2, \dots G$) represent the matrix of Contourlet coefficients of N training images (under different illuminations) of G subjects in the training data at the same scale s and same orientation k . Note that only a subset of the 23 images under different illuminations are used for training. Each column of \mathbf{A}^{sk} contains the Contourlet coefficients of one image. The mean of the matrix is given by

$$\boldsymbol{\mu}^{sk} = \frac{1}{N \times G} \sum_{n=1}^{N \times G} \mathbf{A}_n^{sk} \quad , \quad (1)$$

and the covariance matrix by

$$\mathbf{C}^{sk} = \frac{1}{N \times G} \sum_{n=1}^{N \times G} (\mathbf{A}_n^{sk} - \boldsymbol{\mu}^{sk})(\mathbf{A}_n^{sk} - \boldsymbol{\mu}^{sk})^T \quad . \quad (2)$$

The eigenvectors of \mathbf{C}^{sk} are calculated by Singular Value Decomposition

$$\mathbf{U}^{sk} \mathbf{S}^{sk} (\mathbf{V}^{sk})^T = \mathbf{C}^{sk} \quad , \quad (3)$$

where the matrix \mathbf{U}^{sk} contains the eigenvectors sorted according to the decreasing order of eigenvalues and the diagonal matrix \mathbf{S}^{sk} contains the respective

eigenvalues. Let λ_n (where $n = 1, 2, \dots, N \times G$) represent the eigenvalues in decreasing order. We select the subspace dimension (i.e. number of eigenvectors) so as to retain 90% energy and project the Contourlet coefficients to this subspace. If \mathbf{U}_L^{sk} represents the first L eigenvectors of \mathbf{U}^{sk} then the subspace Contourlet coefficients at scale s and orientation k are given by

$$\mathbf{B}^{sk} = (\mathbf{U}_L^{sk})^T (\mathbf{A}^{sk} - \boldsymbol{\mu}^{sk} \mathbf{p}) \quad , \quad (4)$$

where \mathbf{p} is a row vector of all 1's and equal in dimension to $\boldsymbol{\mu}^{sk}$. Note that \mathbf{U}_L^{sk} represents the subspace for Contourlet coefficients at scale s and orientation k . Similar subspaces are calculated for different scales and orientations using the training data and each time, the subspace dimension is chosen so as to retain 90% energy. In our experiments, we considered three scales and a total of 15 orientations along with the low pass sub-band image. Fig. 3 shows samples of a sub-band image and Contourlet coefficients at two scales and seven orientations.

The subspace Contourlet coefficients were normalized so that the variance along each of the L dimensions becomes equal. This is done by dividing the subspace coefficients by the square root of the respective eigenvalues. The normalized subspace Contourlet coefficients at three scales and 15 orientations of each image are stacked to form a matrix of feature vectors \mathbf{B} where each column is a feature vector of the concatenated subspace Contourlet coefficients of an image. These features are once again projected to a linear subspace however, this time without subtracting the mean. Since the feature dimension is usually large compared to the size of the training data, $\mathbf{B}\mathbf{B}^T$ is very large. Moreover, at most $N \times G - 1$ orthogonal dimensions (eigenvectors and eigenvalues) can be calculated for a training data of size $N \times G$. The $(N \times G)$ th eigenvalue is always zero. Therefore, we calculate the covariance matrix $\mathbf{C} = \mathbf{B}^T \mathbf{B}$ instead and find the $N \times G - 1$ dimensional subspace as follows

$$\mathbf{U}' \mathbf{S} \mathbf{V}^T = \mathbf{C} \quad , \quad (5)$$

$$\mathbf{U} = \mathbf{B} \mathbf{U}' / \sqrt{\text{diag}(\mathbf{S})} \quad . \quad (6)$$

In Eqn. 6, each dimension (i. e. column of $\mathbf{A}\mathbf{U}'$) is divided by the square root of the corresponding eigenvalue so that the eigenvectors in \mathbf{U} (i. e. columns) are of unit magnitude. The last column of $\mathbf{A}\mathbf{U}'$ is ignored to avoid division by zero. Thus \mathbf{U} defines an $N \times G - 1$ dimensional linear subspace. The feature vectors are projected to this subspace and used for classification

$$\mathbf{F} = \mathbf{U}^T \mathbf{B} \quad (7)$$

3 Classification

We tested three different classification approaches. In the first approach, the correlation between the features of the query and the training images was calculated by

$$\gamma = \frac{n \sum \mathbf{t} \mathbf{q} - \sum \mathbf{t} \sum \mathbf{q}}{\sqrt{n \sum (\mathbf{t})^2 - (\sum \mathbf{t})^2} \sqrt{n \sum (\mathbf{q})^2 - (\sum \mathbf{q})^2}} \quad , \quad (8)$$

where \mathbf{t} and \mathbf{q} are the subspace Contourlet coefficients of the target and query faces and n is the subspace dimension. The query image was assigned the identity of the one with the highest correlation. In the second approach, we used the feature to subspace distance for classification and assigned the identity of the nearest subspace to the query face. More specifically, we define face specific subspaces comprising the subspace Contourlet coefficients (i.e. columns of \mathbf{F}) of the face as the basis vectors. This is similar to Lee et al. [13] who defined face specific subspaces using the images as basis vectors. The difference in our case is that the face specific subspace is defined by features rather than the images. In the third classification approach, we train a Support Vector Machine (SVM) [11] using Radial Basis Function (RBF) kernel whose parameters are optimized using the k-fold cross validation approach on the training data. All three classification techniques gave similar identification rates however, the first technique consistently gave much better verification results on all three databases. Therefore, we will report results for classification based on correlation coefficient.

4 Results

Three experiments were performed using our database (4347 images of 106 subjects), the extended Yale B database (1710 images of 38 subjects) and the CMU-PIE database (1344 images of 68 subjects). The number of different illumination conditions for these databases are 23, 45 and 21 respectively. All images were with frontal pose. Details of each experiment are given below.

4.1 Experiment 1

This experiment studies the recognition rate versus the number of subspace Contourlet coefficients. This experiment was first performed using the first session (23 images of 106 subjects) of our database where five images per person were used for training and the rest for testing. The experiment was then repeated by

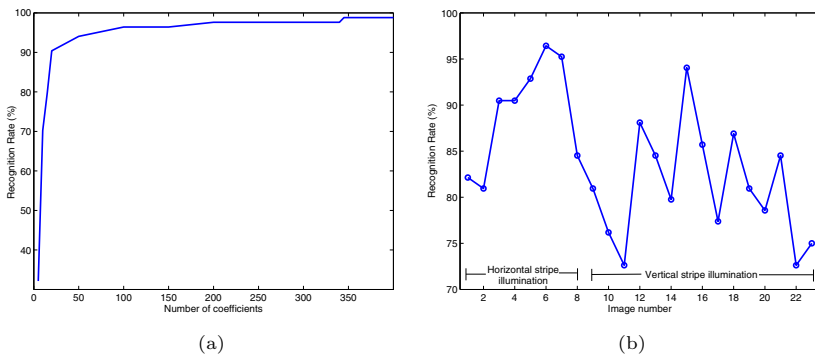


Fig. 4. (a) Recognition rate vs. the number of subspace Contourlet coefficients. (b) Recognition rates for individual images/illumination conditions (x-axis).

training the system with five images per person from the first session and testing with all the images from the second session i.e. 23 images of 83 subjects. Similar results were achieved in both cases. Fig. 4-a shows the plot for the second case. The recognition rate reaches its maximum with only 340 coefficients.

We also studied the relationship of incident light and recognition accuracy and found the recognition rates for individual images corresponding to one of the 23 illumination conditions. The system was trained with a single image per person from the first session and then tested with a single image that corresponds to the same illumination conditions from the second session. Results are reported in Fig. 4-b. As expected, the images with frontal illumination yield high recognition rates. Interestingly, for vertical stripe illumination, the recognition rate first drops and then rises again as the stripe moves away from the center of the screen indicating a non-linear relationship between the recognition accuracy and lateral angle of incident light.

4.2 Experiment 2

In experiment 2, we study the relationship between the number of training images and recognition/verification rates. One or more images/person are used for training and a single image/person is used for testing. We avoid testing all combinations of training images and take advantage of Lee et al.'s [13] findings that one or two frontal and four to five laterally lit images are sufficient for training.

This experiment was performed using the first session of our database, the extended Yale B database and the CMU-PIE database. Fig. 5 shows the recognition and verification rates, using our database, when 5 to 8 images per identity are used for training and the remaining are used for testing. Table 1 summarizes the results. Using 8 training images, the recognition and verification rate at 0.001FAR was 99.87%. The 8 training images that gave the best performance were number 2, 5, 12, 14, 17, 20, 21, 23. Note that this is consistent with the findings in [13]. For fewer training images, we removed images that were lit from large angles one by one in the following order 23, 21, 12, 20, 2, 17, 14 until we

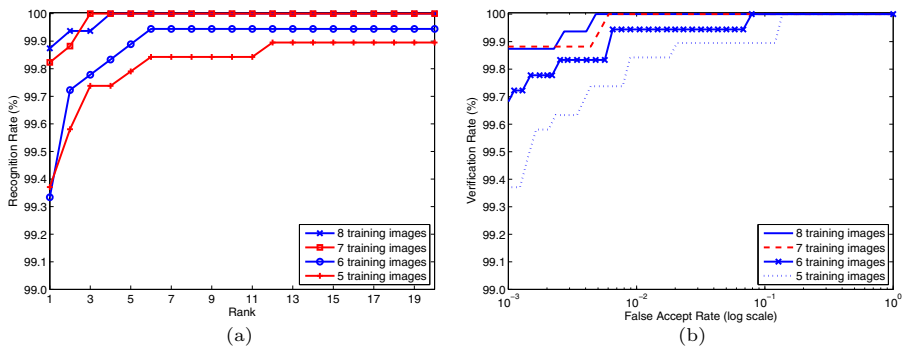
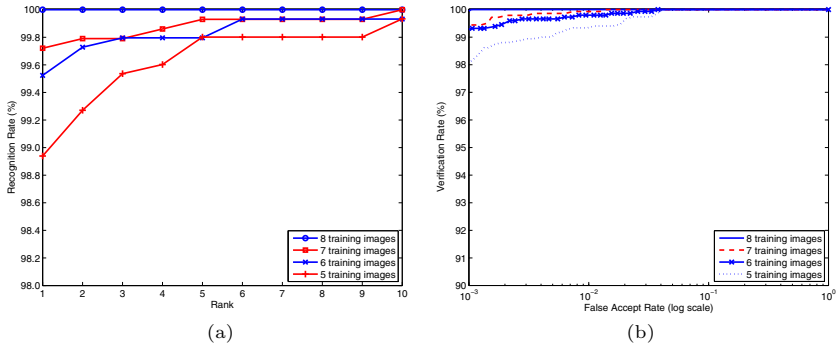


Fig. 5. Experiment 2 results for our database. (a) CMC curves for 5 to 8 training images and the corresponding (b) ROC curves.

Table 1. Experiment 2 results (in %) using our database

Training images	error rate	recog. rate	verif. rate at 0.1% FAR
8	0.13	99.87	99.87
7	0.18	99.82	99.87
6	0.66	99.33	99.67
5	0.63	99.37	99.37
4	0.89	99.11	98.76
3	1.32	98.68	98.76
2	3.10	96.90	95.69
1	14.41	85.59	79.46

**Fig. 6.** Experiment 2 results for the extended Yale B database. (a) CMC curves for 5 to 8 training images and the corresponding (b) ROC curves.

were left with image 5 only. This order was chosen so that there is at least one frontal lit image in the training data and the lateral images are the ones that make a smaller angle with the optical axis. This is sensible from a practical stand point because placing lights at smaller angles requires less space.

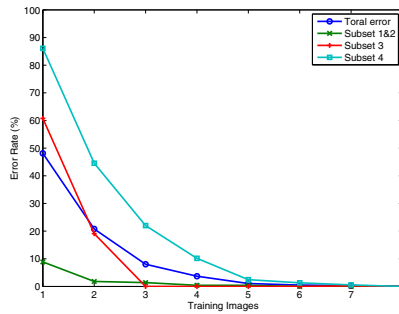
Fig. 6 shows the CMC and ROC curves of our algorithm on the extended Yale B database and a summary of the results is presented in Table 2. Fig. 7 shows plots of error rates for different subsets of the database for direct comparison with [13]. Our algorithm achieved 100% recognition rate and 100% verification rate at 0.001FAR on this database. On the CMU-PIE database [20], with just five images used for training and the remaining for testing, our approach achieved 100% recognition rate and 100% verification rate at 0.01% FAR. Using three training images, the recognition rate dropped to 94.6%.

4.3 Experiment 3

In this experiment, we study the effects of time lapse between training and test images. This experiment cannot be performed on the Yale B and CMU-PIE databases because they were acquired in a single session per subject. Therefore, we perform this experiment on our database only. The setup of this experiment is similar to experiment 2 except that the system is trained using images from

Table 2. Experiment 2 results (in %) using the extended Yale B database

Training images	error rates for subset				recog. rate	verif. rate at 0.1% FAR	FAR at 100% recog. rate
	1&2	3	4	total			
8	0	0	0	0	100	100	0.1
7	0	0.22	0.56	0.28	99.72	99.44	1.39
6	0	0	1.31	0.48	99.52	99.25	3.78
5	0.42	0	2.44	1.06	98.94	98.08	4.16
4	0.42	0	10.15	3.69	96.31	94.24	74.89
3	1.39	0	22.0	8.02	91.98	88.64	98.86
2	1.80	19.08	44.55	20.78	79.22	70.90	98.39
1	8.86	60.75	86.09	48.13	58.17	36.81	98.59

**Fig. 7.** Error rates for different subsets of the extended Yale B database

the first session and then tested on images from the second session. The average time lapse between the first and second sessions in our database was 60 days. The gallery size is 106 and number of test images is $23 \times 83 = 1909$. Fig. 8 shows the CMC and ROC curves for this experiment for different number of training images. The results are summarized in Table 3. The recognition rate drops to 96.65% and the verification rate at 0.001FAR drops to 94.34%. The algorithm performs well for as low as 5 training images and then breaks down.

Table 3. Experiment 3 results (in %) using our database

Training images	error rate	recog. rate	verif. rate at 0.1% FAR
8	3.35	96.65	94.34
7	3.35	96.65	93.29
6	4.06	95.91	92.51
5	6.02	93.98	90.10
4	9.32	90.68	83.55
3	10.58	89.42	77.06
2	15.14	84.86	70.40
1	36.83	63.17	37.45

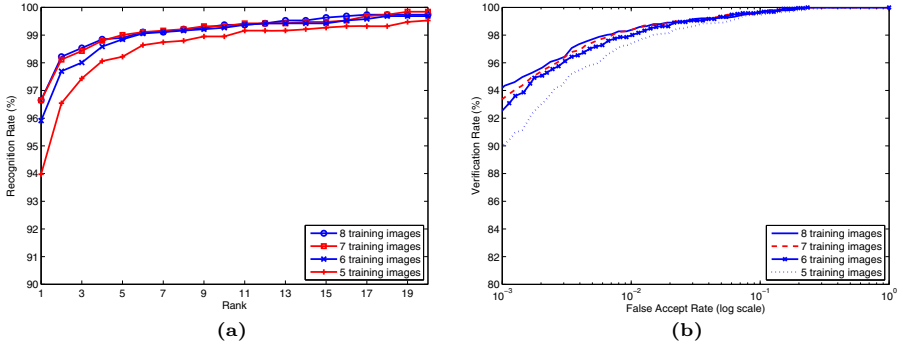


Fig. 8. Experiment 3 results for our database. (a) CMC curves for 5 to 8 training images and the corresponding (b) ROC curves.

Table 4. Comparison of different techniques on the Yale B (10 subjects) and extended Yale B databases. The second citation (if present) refers to the source of results.

Method	subjects	Error rate on Yale B database			
		subset 1&2	subset 3	subset 4	total
Eigen Face w/o first 3 [13]	10	0.0	19.2	66.4	25.8
Cones-attached [9] [13]	10	0.0	0.0	8.6	2.7
Harmonic Image-cast [2] [13]	10	0.0	0.0	2.7	0.85
9 Points of light [13]	10	0.0	0.0	0.0	0.0
Logarithmic Total Variation [6] [21]	38	0.0	1.6	1.1	-
Local Texture Features [21]	38	0.0	0.0	0.8	-
Subspace Contourlet Coeff.	38	0.0	0.0	0.0	0.0

4.4 Timing and Comparison with Other Techniques

Using a Matlab implementation on a 2.4GHz machine with 4GB RAM, the training time using our database of 106 subjects and 6 images per subject was 2 minutes. The recognition time on the same machine and with the same gallery size was 258 msec. The average time required for calculating the Contourlet transform of a face at 3 scales and 15 orientations was 100 msec and for matching two faces was 0.4 msec. Table 4 shows a comparison of our algorithm to existing techniques.

5 Conclusion

We presented a novel algorithm that exploits desktop equipment for face recognition under varying illumination. We demonstrated that it is possible to construct subspaces in the feature space for illumination invariant face recognition using multiple images of the face under extended light source illuminations from a computer screen. Our results on the extended Yale B and CMU-PIE databases revealed that the subspace constructed from the Contourlet coefficients [8] of 5

to 8 images out performs existing state of the art algorithms. In the future, we plan to use our database of images to construct the 3D face models for pose invariant recognition.

Acknowledgments

Thanks to M. Do for the Contourlet Toolbox and R. Owens for the useful discussions. This research is sponsored by ARC grant DP0881813.

References

1. Arandjelovic, O., Cipolla, R.: Face Recognition from Video Using the Generic Shape-Illumination Manifold. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 27–40. Springer, Heidelberg (2006)
2. Basri, R., Jacobs, D.: Lambertian Reflectance and Linear Subspaces. *IEEE Trans. on PAMI* 25(2), 218–233 (2003)
3. Belhumeur, P., Hespanha, J., Kriegman, D.: Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. *IEEE Trans. on PAMI* 19, 711–720 (1997)
4. Belhumeur, P., Kriegman, D.: What Is the Set of Images of an Object Under All Possible Illumination Conditions? *Int'l Journal of Computer Vision* 28(3), 245–260 (1998)
5. Bowyer, K.W., Chang, K., Flynn, P.: A Survey Of Approaches and Challenges in 3D and Multi-modal 3D + 2D Face Recognition. *CVIU* 101, 1–15 (2006)
6. Chen, T., Yin, W., Zhou, X., Comaniciu, D., Huang, T.: Total Variation Models for Variable Lighting Face Recognition. *IEEE Trans. on PAMI* 28(9), 1519–1524 (2006)
7. Chu, R., Liao, S., Zhang, L.: Illumination Invariant Face Recognition Using Near-Infrared Images. *IEEE Trans. on PAMI* 29(4), 627–639 (2007)
8. Do, M.N., Vetterli, M.: The Contourlet Transform: an Efficient Directional Multiresolution Image Representation. *IEEE Trans. on Image Processing* 14(12), 2091–2106 (2005)
9. Georghiades, A., Belhumeur, P., Kriegman, D.: From Few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. on Pattern Analysis and Machine Intell.* 6(23), 643–660 (2001)
10. Hillinan, P.: A Low-Dimensional Representation of Human Faces for Arbitrary Lighting Conditions. In: *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 995–999 (1994)
11. Joachims, T.: Making large-Scale SVM Learning Practical. *Advances in Kernel Methods - Support Vector Learning*. MIT-Press, Cambridge (1999)
12. Lee, K., Kriegman, D.: Online Probabilistic Appearance Manifolds for Video-based Recognition and Tracking. In: *CVPR*, vol. 1, pp. 852–859 (2005)
13. Lee, K., Ho, J., Kriegman, D.: Acquiring Linear Subspaces for Face Recognition under Variable Lighting. *IEEE Trans. on PAMI* 27(5), 684–698 (2005)
14. Li, Y., Gong, S., Liddell, H.: Constructing Facial Identity Surfaces for Recognition. *Int. J. Comput. Vision* 53(1), 71–92 (2003)
15. Liu, C., Wechsler, H.: Face Recognition Using Independent Gabor Wavelet Features. *Audio- and Video-Based Biometric Person Auth.*, 20–25 (2001)

16. Liu, L., Wang, Y., Tan, T.: Online Appearance Model Learning for Video-Based Face Recognition. In: IEEE CVPR, pp. 1–7 (2007)
17. Mian, A.: ‘Shade Face: Multiple Image based 3D Face Recognition. In: IEEE Workshop on 3D Digital Imaging and Modeling, pp. 1833–1839 (2009)
18. Schechner, Y., Nayar, S., Belhumeur, P.: A Theory of Multiplexed Illumination. In: IEEE Int’l Conf. on Computer Vision, pp. 808–815 (2003)
19. Shen, L., Bai, L.: A Review on Gabor Wavelets for Face Recognition. *Pattern Analysis and Appl.* 19, 273–292 (2006)
20. Sim, T., Baker, S., Bsat, M.: The CMU Pose, Illumination, and Expression Database. *IEEE Trans. on PAMI* 25(12), 1615–1618 (2003)
21. Tan, X., Triggs, B.: Enhanced local texture feature sets for face recognition under difficult lighting conditions. In: AMFG (2007)
22. Tangelder, J., Schouten, B.: Learning a Sparse Representation from Multiple Still Images for On-Line Face Recognition in an Unconstrained Environment. In: ICPR, pp. 10867–1090 (2006)
23. Turk, M., Pentland, A.: Eigenfaces for Recognition. *Journal of Cognitive Neuroscience* 3, 71–86 (1991)
24. Viola, P., Jones, M.J.: Robust Real-Time Face Detection. *IJCV* 57(2), 137–154 (2004)
25. Wiskott, L., Fellous, J.M., Kruger, N., Malsgurg, C.: Face Recognition by Elastic Bunch Graph Matching. *IEEE Trans. on PAMI* 19(7), 775–779 (1997)
26. Zhao, W., Chellappa, R., Phillips, P.J., Rosenfeld, A.: Face Recognition: A Literature Survey. *ACM Computing Survey* 35(4), 399–458 (2003)
27. Zhang, H., Zhang, B., Huang, W., Tian, Q.: Gabor Wavelet Associative Memory for Face Recognition. *IEEE Trans. Neural Networks* 16(1), 275–278 (2005)

Shape and Texture Based Plant Leaf Classification

Thibaut Beghin, James S. Cope, Paolo Remagnino, and Sarah Barman

Digital Imaging Research Centre, Kingston University, London, UK
{t.beghin,j.cope,p.remagnino,s.barman}@kingston.ac.uk

Abstract. This article presents a novel method for classification of plants using their leaves. Most plant species have unique leaves which differ from each other by characteristics such as the shape, colour, texture and the margin. The method introduced in this study proposes to use two of these features: the shape and the texture. The shape-based method will extract the contour signature from every leaf and then calculate the dissimilarities between them using the Jeffrey-divergence measure. The orientations of edge gradients will be used to analyse the macro-texture of the leaf. The results of these methods will then be combined using an incremental classification algorithm.

Keywords: Plant identification; Shape-based analysis; texture-based analysis; Sobel operator; incremental classification.

1 Introduction

The role of plants is one of the most important in the natural circle of life. As they form the bulk of the living organisms able to convert the sun light energy into food, they are indispensable to almost every other form of life. They have interested humans since Greek antiquity and the efforts to classify them is, perhaps, the most ancient activity of Science.

Since the development of a systematic classification of plants by the Swedish botanist Carolus Linnaeus in the 18th century [9], plant classification has been attempted in many different ways. The first person who studied the leaf features in this purpose was L.R. Hicher in 1973.

Since then, with the dramatic development of digital image processing, machine vision and pattern recognition, numerous techniques for plant classification using leaves have been investigated. To contribute to these techniques, this paper proposes to develop a classification system using both shape-based and texture-based analysis.

Section 2 introduces the dataset used in this paper, and the outlines the pre-processing performed.

Section 3 presents the shape-based method which uses the contour signatures of the leaves and calculates the dissimilarities between them using the Jeffrey distance. This method has proven its effectiveness for leaf identification [15,13,14,3,19].

The texture-based method is presented in Section 4. The most common techniques of texture description are, in general, based on the statistical analysis of the pixels (co-occurrence matrices, etc.) [8,21,5,17], and their spectral analysis (Fourier Transform, Wavelet Transform, Gabor filters, etc.) [20,11,4,22,12,7,1].

Although there are numerous techniques for texture classification, few of them have been applied to leaves [6,10,18,2]. The technique implemented by the authors makes use of the Sobel operator to analyse the macro-texture of the leaf.

Finally, Section 5, will present an incremental algorithm used to combine the results of the previous methods using probability density functions.

2 Data Pre-processing

The leaves used in this work were collected in the Royal Botanic Gardens, Kew, UK. The dataset contains 3 to 10 leaves from each of 18 different species.

As the colour of the leaves cannot be used as reliable information, since it varies depending on the period of the year as well as other factors, the data has been transformed into greyscale images. The image background, the paper on which the leaf is mounted, is removed using Otsu's thresholding method [16].

3 Analysis of the Contour Signature

Two contour signatures are calculated for analysing leaf shapes. For each leaf, first the outline is extracted by selecting from the image the foreground pixels which neighbour a background pixel on at least one of their four main sides (N,S,E,W). Moving in a clockwise direction, for every $\frac{l}{n}$ th contour pixel, where l is the length of the outline and n is the number of points to be sampled, two values, $f(i)$ and $g(i)$ are calculated:

$$f(i) = \sqrt{(cont_x(j) - cent_x)^2 + (cont_y(j) - cent_y)^2} \quad (1)$$

$$g(i) = \left| \tan\left(\frac{cont_x(j) - cent_x}{cont_y(j) - cent_y}\right) - \frac{2i\pi}{n} \right| \quad (2)$$

Where, $j = \frac{i \times l}{n}$, $cont_x(j), cont_y(j)$ are the x and y co-ordinates respectively for the j th contour pixel, and $cent_x, cent_y$ are the x and y co-ordinates of the leaf's centroid.

The first of the resulting signatures f , gives the distances between the contour point and the centre of the leaf. The second, g , is the absolute difference between the angle at the leaf centre between the starting point and the current point, and the corresponding angle on a circle. Together, these two signatures provide a significant amount of information about the leaf's shape.

These signatures are treated like probability density functions (pdfs) by dividing each value by the sum of all the values in the signature. Doing this provides us with scale-invariance. The difference between the signatures for two leaves

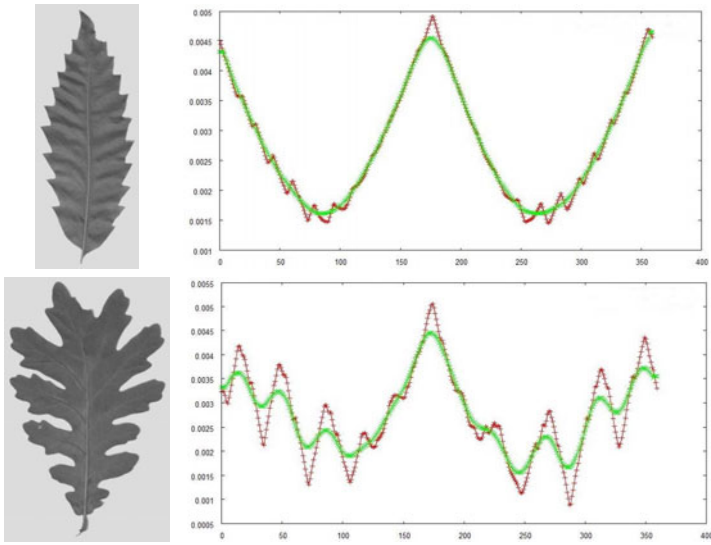


Fig. 1. With the smoothing, the lobed leaf (bottom) is distinguished from the serrated leaf (top)

can then calculated using the Jeffrey-divergence distance measure. For two pdfs, f_a and f_b , the distance between them, $JD(f_a, f_b)$, is calculated as follows:

$$JD(f_a, f_b) = \sum_i \sum_j f_a(i, j) \log\left(\frac{2f_a(i, j)}{f_a(i, j) + f_b(i, j)}\right) + f_b(i, j) \log\left(\frac{2f_b(i, j)}{f_a(i, j) + f_b(i, j)}\right) \tag{3}$$

Since the signatures for two leaves may begin at different points on the leaves, the signature must be aligned before they can be compared. This can by using

Table 1. The confusion matrix for the contour signatures, including lobe differentiation

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
0	0.94	0	0	0	0	0	0	0	0	0.05	0	0	0	0	0	0	0	0	
1	0	0.64	0	0	0.08	0.24	0	0.04	0	0	0	0	0	0	0	0	0	0	
2	0	0	0.47	0.11	0	0	0.19	0	0	0	0	0	0	0.19	0	0.02	0	0	
3	0	0	0	0.56	0	0	0.43	0	0	0	0	0	0	0	0	0	0	0	
4	0	0.37	0	0	0.37	0.18	0	0.06	0	0	0	0	0	0	0	0	0	0	
5	0	0	0	0	0.18	0.78	0	0	0.06	0	0	0	0	0	0	0	0	0	
6	0	0	0	0.36	0	0	0.64	0	0	0	0	0	0	0	0	0	0	0	
7	0	0.22	0	0	0.13	0.19	0	0.38	0.02	0	0	0.02	0	0	0	0	0	0	
8	0	0.04	0	0	0.04	0.28	0	0.08	0.40	0	0	0.16	0	0	0	0	0	0	
9	0	0	0	0	0	0	0	0	0	0.97	0	0	0	0	0	0	0	0.02	
10	0	0	0	0	0	0	0	0	0	0	1.00	0	0	0	0	0	0	0	
11	0	0.06	0	0	0.07	0.20	0	0.11	0.08	0	0	0.45	0	0	0	0	0	0	
12	0	0	0	0	0	0	0	0	0	0	0	0	0.77	0	0.22	0	0	0	
13	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00	0	0	0	0	
14	0	0	0	0	0	0	0	0	0	0.11	0	0	0.22	0	0.44	0	0.22	0	
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00	0	0	
16	0	0	0	0	0	0	0	0	0	0.05	0	0	0	0	0	0.02	0	0.91	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0.22	0	0	0	0	0.77

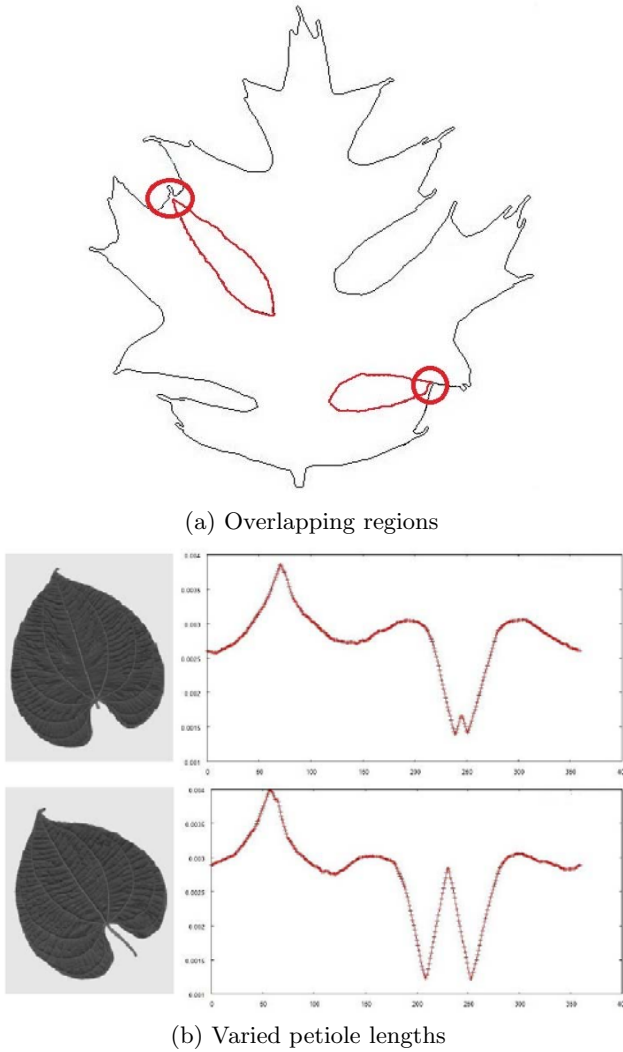


Fig. 2. The main issues for the contour signature method

cross-correlation, whereby the amount to offset the second leaf’s signature by is calculated as follows:

$$offset = \underset{j=0..(n-1)}{argmin} \left(\sum_{i=0}^n (f_a(i) - f_b(i + j))^2 \right) \tag{4}$$

3.1 Differentiation between Lobed and Unlobed Leaves

Shape-based leaf classification can be improved by differentiating between lobed and unlobed leaves. This can be done by calculating the number of inflection

points in the contour distance signature. Each point in the signature is compared to the 3 points either side of it. If the point is either less than all these neighbours, or greater than them, then the point is an inflection point. Once every inflection point has been detected, they are counted and if the total number is above some threshold, the leaf is considered lobed.

Using this method, serrated leaves would be identified as having many lobes. To prevent this, the signature is first smoothed by using a Gaussian filter. The difference between a lobed and a serrated leaf, as well as their contour graphs (normal and smoothed), can be observed in figure 1. The normal graph would give a lot of inflection points for these two leaves and would classify both in the lobed category although only the first one actually is.

3.2 Results

The results of the contour signature method can be seen in table 1. All the leaves in the dataset were compared to all others, and classified as the same species as the closest other leaf. The overall correct classification rate is 69.2%. Whilst some of the species achieved a high recognition rate (with 3 at 100%), many did much worse, with 6 under 50%. Part of reason for this is the high intra-species variation present within some lobed species, and the low inter-species variation between species with ovate leaves. Another cause of errors appears when leaves have overlapping regions, which cause the contours to be incorrectly traced, as shown in figure 2a. Figure 2b shows that petiole (stems) cut that different lengths before imaging the leaves can also cause problems.

4 Texture Analysis Using Sobel

The results for the contour signatures suggest that leaves cannot be adequately classified based on shape alone. The texture is also an important feature of the leaf. Two types of texture can be defined: the micro-texture at the microscopic scale and the macro-texture which is the pattern formed by the venation of the leaf. The venation is specific to every leaf, similar to a fingerprint. In this chapter, the concept of macro-texture is quantified using edge gradients.

4.1 Histogram of the Gradient Intensity

For each image, we calculate a histogram of the gradient orientations, whereby for the angle θ :

$$h(\theta) = \sum_x \sum_y M(x, y) \text{ if } \Theta(x, y) = \theta, 0 \text{ otherwise} \quad (5)$$

Where $M(x, y)$ is the gradient magnitude at pixel (x, y) and $\Theta(x, y)$ is the gradient direction, calculated using the Sobel operator. This histogram provides a description of the relative directions of the main veins. Examples of these histograms for four leaves from the species *Quercus Ilex* can be seen in figure 3.

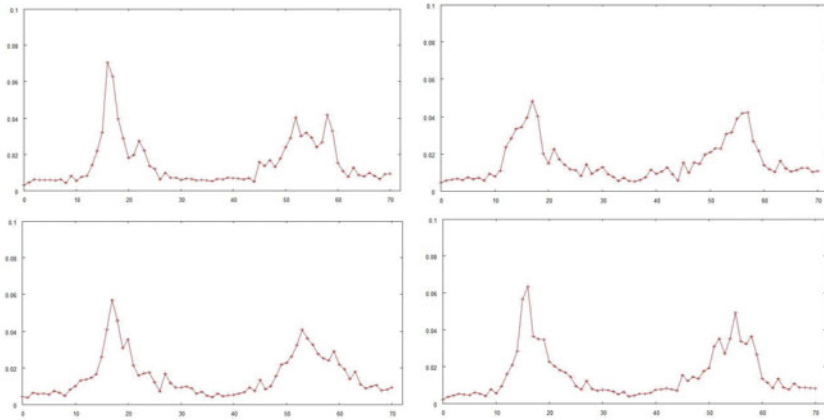


Fig. 3. Sobel direction histograms for four leaves from the same species

The difference between the gradient histograms is again calculated using the Jeffrey-divergence distance measure. The confusion matrix for this method can be seen in table 2. Table 3 shows the correct classification rates for the shape and texture methods. Whilst the Sobel method only achieved a rate of 66.1%, it can be seen that though some species are classified more accurately using the contour method, others do much better using the Sobel method. For instance, the *Agrifolia*, the 1982 and the 1998-4292 are well recognized by the contour method, due to low intra-species variation, and very badly by the Sobel method, possibly due to uneven lighting in the images. On the other hand, the *Ellipsoidalis*, the *Turneri* and the 2005 are better identified by the Sobel method, where flatter leaves created less shadowing. It may therefore be possible to greatly improve the overall results by combining the two methods in the correct manner.

Table 2. The confusion matrix for the gradient histograms

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	0.27	0.11	0	0	0	0	0	0	0.13	0	0	0	0.13	0	0	0	0.33	0
1	0	0.88	0	0	0.12	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0.77	0.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0.25	0.56	0	0	0.18	0	0	0	0	0	0	0	0	0	0	0
4	0	0.37	0	0	0.43	0.12	0	0	0.06	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0.06	0.81	0	0	0.12	0	0	0	0	0	0	0	0	0
6	0	0	0	0.28	0	0	0.72	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0.02	0	0.72	0.25	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0.04	0	0.08	0.76	0	0	0.04	0	0	0	0	0	0.08
9	0	0.03	0	0	0	0	0	0	0	0.32	0	0.11	0.12	0	0.16	0	0.26	0
10	0	0	0	0	0	0.06	0	0.06	0	0	0.75	0	0	0	0	0	0.12	0
11	0	0.03	0	0	0	0	0	0.01	0.02	0.17	0	0.29	0.11	0	0.14	0	0.19	0
12	0	0	0	0	0	0	0	0	0	0	0	0	1.00	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0.48	0	0.52	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0.11	0	0.88	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0.06	0	0.93	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.66	0	0.33

Table 3. Results for the two methods

		Contour Score	Sobel Score
1	Agrifolia	0.94	0.27
2	Castaneifolia	0.64	0.88
3	Ellipsoidalis	0.47	0.77
4	Frainetto	0.56	0.56
5	Hispanica	0.37	0.43
6	Ilex	0.78	0.81
7	Robur	0.64	0.72
8	Turneri	0.38	0.72
9	Variabilis	0.40	0.76
10	1982	0.97	0.32
11	1995	1.00	0.75
12	1996	0.45	0.29
13	1998-523	0.77	1.00
14	1998-4292	1.00	0.48
15	2005	0.44	0.88
16	2008	1.00	0.93
17	F184	0.91	1.00
18	Passifloranono	0.77	0.33

Table 4. The confusion matrix for the final, incremental classification

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	1.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0.88	0	0	0.12	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0.86	0.13	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0.81	0	0	0.18	0	0	0	0	0	0	0	0	0	0	0
4	0	0.37	0	0	0.43	0.18	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0.37	0.56	0	0	0.06	0	0	0	0	0	0	0	0	0
6	0	0	0	0.24	0	0	0.76	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0.02	0	0.75	0.22	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0.04	0	0.16	0.76	0	0	0.04	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0.50	0	0	0	0	0	0	0.50	0
10	0	0	0	0	0	0	0	0	0	0	1.00	0	0	0	0	0	0	0
11	0	0.27	0	0	0	0.06	0	0.07	0.18	0	0	0.40	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	1.00	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0.11	0	0.88	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00

5 Incremental Classification

It seems that leaves cannot be sufficiently well classified based on the shape or the texture alone, though good results may be achieved by using both of these features. In order to limit the risks of failure and improve the recognition rate, we will use an incremental classification method. Firstly, the calculation of the inflection points is used to separate the lobed and unlobed leaves. The species which are in the same category as the leaf being analysed are kept and the other species are ignored.

Secondly, a classification using only the contour signature method is performed (the shape of the leaf being the most important feature for classification). Leaves for which the distance between their contour signatures and those of the leaf being classified are greater than some threshold are removed. The same procedure is then performed on the remaining leaves using the texture histograms.

For the final remaining leaves, the distances between both contour signature and the texture histogram are combined, and the leaf is classified as the same species as the closest of these. The results for this are shown in table 4. The overall classification rate is 81.1%, a clear improvement over the separate methods.

6 Conclusion

In this work, an efficient classification framework was proposed to classify a dataset of 18 species of leaves.

Firstly, a classification based on the shape of the leaf is described. Two contour signatures are calculated based on the distance and angle of contour points from the leaf's centre. This operation is done for every leaf of the dataset and the dissimilarities between the graphs are calculated using the Jeffrey distance. This classification, called the contour signature method, presents quite good results. Further improvement is made by the separation of the lobed leaves from the unlobed leaves by the calculation of the signature's inflection points.

Secondly, a classification using the Sobel operator is used in order to capture the dissimilarities of the macro-texture of the leaves. A histogram is formed from the orientation and magnitude of the edge gradients. Finally, a method combining the lobe differentiation, the shaped-based and the texture-based method through the use of probability density functions is implemented. The incremental process is intended to extract the most potential from each individual method. The results show that 10 species out of 18 are successfully classified with a classification rate greater than 85% and 4 with one of more than 75%. The overall classification rate was 81.1%.

The identification of the leaves is a difficult problem because there is often high intra-species variability, and low inter-species variation. Nevertheless, the approach adopted in this work demonstrates the classification of leaves using a combination of relatively simple methods is a valid and promising approach.

References

1. Arivazhagan, S., Ganesan, L., Priyal, S.P.: Texture classification using Gabor wavelets based rotation invariant features. *Pattern Recognition Letters* 27, 1976–1982 (2006)
2. Casanova, D., de Mesquita Sá Jr., J.J., Bruno, O.M.: Plant leaf identification using Gabor wavelets. *International Journal Of Imaging Systems And Technology* 19, 236–243 (2009)
3. Du, J.X., Wang, X.F., Zhang, G.J.: Leaf shape based plant species recognition. *Applied Mathematics and Computation* 185, 883–893 (2007)

4. Gibson, D., Gaydecki, P.A.: Definition and application of a Fourier domain texture measure: application to histological image segmentation. *Computers in Biology and Medicine* 25, 551–557 (1995)
5. Gotlieb, Calvin, C., Kreyszig, H.E.: Texture descriptors based on co-occurrence matrices. *Computer Vision, Graphics, And Image Processing* 51, 70–86 (1990)
6. Gu, X., Du, J.X., Wang, X.F.: Leaf recognition based on the combination of wavelet transform and gaussian interpolation. In: Huang, D.-S., Zhang, X.-P., Huang, G.-B. (eds.) *ICIC 2005*. LNCS, vol. 3644, pp. 253–262. Springer, Heidelberg (2005)
7. Haley, G.M., Manjunath, B.S.: Rotation-invariant texture classification using a complete space-frequency model. *IEEE Trans. Image Processing* 8, 255–269 (1999)
8. Kashyap, R.L., Chellappa, R., Ahuja, N.: Decision rules for choice of neighbors in random field models of images. *Computer Graphics and Image Processing* 15, 301–318 (1981)
9. Linnaei, C.: *Systema Naturae Per Regna Tria Naturae* (1758-1759)
10. Liu, J., Zhang, S., Deng, S.: A method of plant classification based on wavelet transforms and support vector machines. In: Huang, D.-S., Jo, K.-H., Lee, H.-H., Kang, H.-J., Bevilacqua, V. (eds.) *ICIC 2009*. LNCS, vol. 5754, pp. 253–260. Springer, Heidelberg (2009)
11. Mallat, S.G.: A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. Pattern Analysis and Machine Intelligence* 11, 674–693 (1989)
12. Manthalkar, R., Biswas, P.K., Chatterji, B.N.: Rotation and scale invariant texture features using discrete wavelet packet transform. *Pattern Recognition Letters* 24, 2455–2462 (2003)
13. Mokhtarian, F., Abbasi, S.: Matching shapes with self-intersection: application to leaf classification. *IEEE Trans. Image Processing* 13, 653–661 (2004)
14. Neto, J., Meyer, G., Jones, D., Samal, A.: Plant species identification using elliptic Fourier leaf shape analysis. *Computers And Electronics In Agriculture* 50, 121–134 (2006)
15. Oide, M., Ninomiya, S.: Discrimination of soybean leaflet shape by neural networks with image input. *Computers And Electronics In Agriculture* 29, 59–72 (2000)
16. Otsu, N.: A threshold selection method from gray level histograms. *IEEE Trans. Systems, Man and Cybernetics* 9, 62–66 (1979)
17. Partio, M., Cramariuc, B., Gabbouj, M.: An ordinal co-occurrence matrix framework for texture retrieval. *Image and Video Processing* (2007)
18. Ramos, E., Fernandez, D.S.: Classification of leaf epidermis microphotographs using texture features. *Ecological Informatics* 4, 177–181 (2009)
19. Tak, Y.S., Hwang, E.: Pruning and matching scheme for rotation invariant leaf image retrieval. *KSII Trans. Internet And Information Systems* 2, 280–298 (2008)
20. Turner, M.: Texture discrimination by Gabor functions. *Biological Cybernetics* 55, 71–82 (1986)
21. Unser, M.: Local linear transform for texture measurements. *Signal Processing* 11, 61–79 (1986)
22. Unser, M.: Texture classification and segmentation using wavelet frames. *IEEE Trans. Image Processing* 11, 1549–1560 (1995)

A New Approach of GPU Accelerated Visual Tracking

Chuantao Zang and Koichi Hashimoto

Graduate School of Information Sciences, Tohoku University
6-6-01 Aramaki Aza Aoba, Aoba-ku, Sendai 980-8579, Japan
{chuantao,koichi}@ic.is.tohoku.ac.jp

Abstract. In this paper a fast and robust visual tracking approach based on GPU acceleration is proposed. It is an effective combination of two GPU-accelerated algorithms. One is a GPU accelerated visual tracking algorithm based on the Efficient Second-order Minimization (GPU-ESM) algorithm. The other is a GPU based Scale Invariant Feature Transform (SIFT) algorithm, which is used in those extreme cases for GPU-ESM tracking algorithm, i.e. large image differences, occlusions etc. System performances have been greatly improved by our combination approach. We have extended the tracking region from a planar region to a 3D region. Translation details of both GPU algorithms and their combination strategy are described. System performances are evaluated with experimental data. Optimization techniques are presented as a reference for GPU application developers.

Keywords: ESM, SIFT, GPU, Visual tracking.

1 Introduction

Visual tracking is the critical task in computer vision applications, such as visual servo, augmented reality, etc. Visual tracking methods can be mainly divided into two categories: Feature-based methods and Region-based methods[1]. Feature-based methods mainly track local features such as points, line segments, edges or corners in the images. These local feature detections are easy to process but sensitive to illumination change, occlusion and so on. Region-based methods only use the image intensity information in a certain region. By minimizing the sum of squared differences (SSD) between a region in reference image and a warped region in current image, the transformation parameters can be estimated[2] in these methods. For example, the transformation between two images of a plane is a homography[3]. A well-known Region-based method is the Lucas-Kanade algorithm [4][5]. It computes the displacement of points between consecutive frames when the image brightness constancy constraint is satisfied.

To minimizing the SSD between a template region and a warped region in Lucas-Kanade algorithm, many nonlinear optimization approaches have been proposed with different kinds of approximations, such as Standard Newton method [3], Gauss-Newton approximation. Among these solutions, the Efficient Second-order Minimization (ESM) algorithm[6] is an elegant idea which obtains the same convergence speed as standard Newton method while not computing the computationally costly Hessian matrix. Based on the ESM algorithm, Malis has proposed an efficient “ESM visual tracking algorithm” and extended it to visual servo[6][7].

However, when considering a visual tracking system, the main requirements of the tracking algorithms are about efficiency, accuracy and stability. From our experience, with the increase of tracking region size (for example a 360×360 pixels region), the ESM computation still costs too much time and induces a relative low processing speed. This low processing speed will cause a larger image difference in the two successive images of a fast moving object. As ESM tracking algorithm can only work well with small image differences, these large differences will cause tracking failure.

To deal with these problems, we propose a novel approach of using GPU as coprocessor to enhance the system performance. Our contributions are mainly as follows.

We present a GPU based ESM tracking algorithm (GPU-ESM tracking) to address the need for faster tracking algorithms. The speedup allows for a higher speed camera so that there will be smaller difference between two successive frames, which will make the ESM tracking result more reliable and robust. Besides GPU-ESM, we adopt GPU based object recognition algorithms to solve those extreme cases for GPU-ESM tracking, such as large image differences and occlusions, etc. We implement Lowe's Scale Invariant Feature Transform (SIFT) algorithm [8] on GPU (GPU-SIFT) and extend GPU-SIFT algorithm with "RANDOM Sample Consensus" (RANSAC) method to increase its accuracy. With an approximately 20 times GPU speedup, our extended GPU-SIFT tracking greatly enhances the system reliability.

We propose an effective combination strategy of both algorithms mentioned above. When GPU-ESM tracking failure happens, GPU-ESM will automatically load the result from GPU-SIFT so that it can continue tracking. Therefore, the whole system can work smoothly with high reliability at a high processing speed. The previous paper [9] mentions the ESM tracking and visual servo and in this paper 3D region tracking is developed with this combination strategy.

The rest of this paper is organized as follows. Section II reviews the relative works on ESM tracking and SIFT algorithms. Section III introduces the translation details of two GPU algorithms so as to fully utilize the parallel capacity of GPU. This part also covers the combination model of both algorithms in detail. Section IV describes the experimental results to validate our proposed approach. Section V describes the key optimization techniques in our GPU applications. Section VI concludes this paper.

2 Related Works

Our proposed approach is a combination of GPU-ESM tracking and GPU-SIFT algorithms. For simplicity, we review the ESM tracking algorithm and SIFT algorithm.

2.1 ESM Tracking Algorithm

ESM tracking algorithm was proposed by Malis in 2004 [6]. By performing second order approximation of the minimization problem with only first order derivative, ESM algorithm can get a high convergence rate and avoid local minima close to the right global minima. Different kinds of its applications have been realized, such as visual tracking of planar object and deformable object [10], visual servo [7] etc.

Suppose the tracking object is planar and projected in a reference image I^* with a "Template" region of m pixels. Tracking this region consists in finding the homography

\mathbf{G} that transforms each pixel P_i^* of the template region into its corresponding pixel in the current image I , i.e. finding the homography \mathbf{G} such that $\forall i \in \{1, 2, \dots, m\}$:

$$I(w(\mathbf{G})(P_i^*)) = I^*(P_i^*) \tag{1}$$

$P_i^* = [u^* \ v^* \ 1]^T$ is the homogeneous image coordinate. Homography \mathbf{G} is defined in the Special Linear group $\mathbb{SL}(3)$. The matrix \mathbf{G} defines a projective transformation in the image. w is a group action defined from $\mathbb{SL}(3)$ on \mathbb{P}^2 :

$$w : \mathbb{SL}(3) \times \mathbb{P}^2 = \mathbb{P}^2 \tag{2}$$

Therefore, for all $\mathbf{G} \in \mathbb{SL}(3)$, $w(\mathbf{G})$ is a \mathbb{P}^2 automorphism:

$$\begin{aligned} w(\mathbf{G}) : \mathbb{P}^2 &\rightarrow \mathbb{P}^2 \\ P^* &\rightarrow P = w(\mathbf{G})(P^*) \end{aligned} \tag{3}$$

such that:

$$P = w(\mathbf{G})(P^*) = \begin{bmatrix} \frac{g_{11}u^* + g_{12}v^* + g_{13}}{g_{31}u^* + g_{32}v^* + g_{33}} \\ \frac{g_{21}u^* + g_{22}v^* + g_{23}}{g_{31}u^* + g_{32}v^* + g_{33}} \\ 1 \end{bmatrix} \tag{4}$$

Suppose that we have an approximation $\widehat{\mathbf{G}}$ of \mathbf{G} , the problem consists in finding an incremental transformation $\Delta\mathbf{G}$, such that the difference between a region in current image I (transformed from Template region by the composition $w(\widehat{\mathbf{G}}) \circ w(\Delta\mathbf{G})$) and the corresponding region in reference image I^* is null.

Homography \mathbf{G} is in the $\mathbb{SL}(3)$ group which is a Lie group. The Lie algebra associated to this group is $SL(3)$. Let $\{\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_8\}$ be a basis of the Lie algebra $SL(3)$. Then a matrix $\mathbf{A}(\mathbf{x})$ can be expressed as follows:

$$\mathbf{A}(\mathbf{x}) = \sum_{i=1}^8 x_i \mathbf{A}_i \tag{5}$$

A projective transformation $\mathbf{G}(\mathbf{x}) \in \mathbb{SL}(3)$ in the neighborhood of \mathbf{I} can be parameterized as follows:

$$\mathbf{G}(\mathbf{x}) = \exp(\mathbf{A}(\mathbf{x})) = \sum_{i=0}^{\infty} \frac{1}{i!} (\mathbf{A}(\mathbf{x}))^i \tag{6}$$

As incremental transformation $\Delta\mathbf{G}$ also belongs to $\mathbb{SL}(3)$, it can be expressed as $\Delta\mathbf{G}(\mathbf{x})$, where \mathbf{x} is a 8×1 vector. Therefore tracking consists in finding a vector \mathbf{x} such that $\forall i \in \{1, 2, \dots, m\}$, the image difference

$$d_i(\mathbf{x}) = I((w(\widehat{\mathbf{G}}) \circ w(\Delta\mathbf{G}(\mathbf{x}))(P_i^*)) - I^*(P_i^*) = 0 \tag{7}$$

Let $\mathbf{d}(\mathbf{x}) = [d_1(\mathbf{x}), d_2(\mathbf{x}), \dots, d_m(\mathbf{x})]^T$ be the $m \times 1$ vector containing the image differences. Therefore, the problem consists in finding $\mathbf{x} = \mathbf{x}_0$ verifying:

$$\mathbf{d}(\mathbf{x}_0) = \mathbf{0} \tag{8}$$

Linearize the vector $\mathbf{d}(\mathbf{x})$ at $\mathbf{x} = \mathbf{0}$ with a second-order Taylor series approximation:

$$\mathbf{d}(\mathbf{x}) = \mathbf{d}(\mathbf{0}) + \mathbf{J}(\mathbf{0})\mathbf{x} + \frac{1}{2}\mathbf{x}^\top \mathbf{H}(\mathbf{0})\mathbf{x} \quad (9)$$

where $\mathbf{J}(\mathbf{0})$ and $\mathbf{H}(\mathbf{0})$ are the Jacobian matrix and Hessian matrix at $\mathbf{x} = \mathbf{0}$, separately. In the ESM algorithm, the Hessian matrix is replaced by a first-order Taylor Series approximation of vector $\mathbf{J}(\mathbf{x})$ about $\mathbf{x} = \mathbf{0}$:

$$\mathbf{J}(\mathbf{x}) = \mathbf{J}(\mathbf{0}) + \mathbf{x}^\top \mathbf{H}(\mathbf{0}) \quad (10)$$

Then Eq. 9 becomes

$$\mathbf{d}(\mathbf{x}) = \mathbf{d}(\mathbf{0}) + \frac{1}{2}(\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x}))\mathbf{x} \quad (11)$$

With some mathematical proofs by Malis [7], Eq. 11 can be simplified to follows:

$$\mathbf{d}(\mathbf{x}_0) = \mathbf{d}(\mathbf{0}) + \mathbf{J}_{\text{esm}}\mathbf{x}_0 = \mathbf{0} \quad (12)$$

where $\mathbf{J}_{\text{esm}} = \frac{1}{2}(\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x}_0))$. Therefore, the solution \mathbf{x}_0 can be obtained by:

$$\mathbf{x}_0 = -\mathbf{J}_{\text{esm}}^+ \mathbf{d}(\mathbf{0}) \quad (13)$$

where $\mathbf{J}_{\text{esm}}^+$ is the pseudoinverse inverse matrix of \mathbf{J}_{esm} . The incremental transformation $\Delta \mathbf{G}(\mathbf{x}_0)$ can be calculated with \mathbf{x}_0 by Eq. 5 and Eq. 6. Then a new homography \mathbf{G} solution can be obtained by such update:

$$\mathbf{G} = \hat{\mathbf{G}}\Delta \mathbf{G}(\mathbf{x}_0) \quad (14)$$

With this homography \mathbf{G} , visual tracking can be implemented.

2.2 SIFT Algorithm

SIFT was proposed by Lowe in 1999 [11]. Compared with other feature matching approaches, the SIFT algorithm has been demonstrated to have a better performance with respect to variations in scale, rotation, and translation [8]. However, its computation involves a high dimensional descriptor which is computational intensive and difficult to apply for realtime processing. To improve its performance, various algorithms have been proposed, including Affine SIFT [12], PCA-SIFT [13] and SURF [14] and so on.

2.3 GPU-Based Visual Tracking

The increasing programmability and computational capability of the GPU has shown great potential for computer vision algorithms which can be parallelized [15]. For example, a versatile framework for programming GPU-based computer vision tasks (radial distortion, corner detection etc) was recently introduced by [15] [16]. There are also GPU implementations for visual tracking, such as GPU-KLT tracker [17], GPU-SIFT [17] [18] [19]. These applications can get a 10~20 times speedup.

3 Implementation

3.1 System Configuration

GPU implementations are realized on a desktop with Intel Core i7-920 (2.67 GHz), 3GB RAM and a NVIDIA GTX295 graphic board. The GTX295 graphic board integrates two GTX280 GPUs inside and has 896MB GPU RAM for each GPU. Operating system is Windows XP (service pack 2).

Our GPU applications are developed with NVIDIA's CUDA (Compute Capability 1.3). In CUDA's programming model, functions are expressed as kernels and the smallest execution unit on GPU is a thread. Usually multiple CUDA kernels are needed to realize different kinds of functions in one algorithm.

3.2 Implementation of GPU-ESM

The proposed GPU-ESM tracking algorithm can be categorized into 6 CUDA kernels.

- 1) Warping. This kernel completes the task that warps a reference image to the current image with a known homography.
- 2) Gradient. This kernel calculates the intensity gradient in X and Y directions.
- 3) Jesm. This kernel calculates the \mathbf{J}_{esm} matrix in ESM algorithm.
- 4) Solving. This kernel finds the solution \mathbf{x} of linear equations

$$\mathbf{J}_{\text{esm}}\mathbf{x} = -\mathbf{d}(\mathbf{0}) \quad (15)$$

\mathbf{J}_{esm} is of $m \times 8$, \mathbf{x} is of 8×1 , therefore this equation is overdetermined. To solve this equation, we multiply the transpose of \mathbf{J}_{esm} on both sides:

$$\mathbf{J}_{\text{esm}}^T \mathbf{J}_{\text{esm}} \mathbf{x} = -\mathbf{J}_{\text{esm}}^T \mathbf{d}(\mathbf{0}) \quad (16)$$

and adopt the Cholesky decomposition method to solve Eq. 16 for the solution \mathbf{x}_0 .

5) Updating. This kernel updates homography with solution \mathbf{x} from "Solving" kernel. Calculation methods are shown in Eq. 5, 6 and 14. In our application, we use the following $SL(3)$ basis matrices:

$$\begin{aligned} \mathbf{A}_1 &= \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \mathbf{A}_2 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} & \mathbf{A}_3 &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \mathbf{A}_4 &= \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ \mathbf{A}_5 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \mathbf{A}_6 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \mathbf{A}_7 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} & \mathbf{A}_8 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \end{aligned} \quad (17)$$

We adopt such approximation to calculate the matrix exponential of $\exp((\mathbf{A}(\mathbf{x}_0)))$:

$$\begin{aligned} \mathbf{G}(\mathbf{x}_0) &= \exp(\mathbf{A}(\mathbf{x}_0)) = \sum_{i=0}^{\infty} \frac{1}{i!} (\mathbf{A}(\mathbf{x}_0))^i \\ &\approx \mathbf{I} + \mathbf{A}(\mathbf{x}_0) + \frac{1}{2} \mathbf{A}(\mathbf{x}_0)^2 + \frac{1}{6} \mathbf{A}(\mathbf{x}_0)^3 \end{aligned} \quad (18)$$

Due to the small eigenvalues of $\mathbf{A}(\mathbf{x}_0)$ (near 0), above approximation can work well without losing accuracy.

6) Correlation. This kernel calculates the correlation of warped region I and template region I^* (m pixels) with the Zero mean Normalized Cross Correlation (ZNCC):

$$\frac{\sum_{k=1}^m (I(k) - \bar{I})(I^*(k) - \bar{I}^*)}{\sqrt{\sum_{k=1}^m (I(k) - \bar{I})^2 \sum_{k=1}^m (I^*(k) - \bar{I}^*)^2}} \quad (19)$$

where \bar{I} and \bar{I}^* are the mean intensity values of warped region I and template region I^* , respectively. As a quality evaluation criterion, the correlation has played two important roles in our application. On one hand, if the correlation is smaller than a preset lower threshold, it will be treated as ESM tracking failure has happened. On the other hand, if the correlation is larger than a preset upper threshold, the iterative ESM processing loop will stop and continue to process the next input image. As an iterative minimization method, such threshold to stop the ESM loop is necessary.

3.3 Implementation of GPU-SIFT

In our GPU-SIFT, we transfer Changchang Wu's GPU-SFIT matching[19] to match the features. Then we adopt the RANSAC method to improve the homography accuracy. RANSAC method has shown a better performance than least squares methods as it can effectively remove some of the mismatched pairs of points in GPU-SIFT.

3.4 Combination Strategy

As mentioned above, ESM tracking algorithm can provide a fast and accurate homography solution when the solution is near the global minimum point, but its convergence region is small. For large image difference it will lose tracking. Meanwhile, SIFT algorithm can offer a robust solution in a large region. But it is not fast enough for a real time visual servo system. Limited by the mismatched outliers, the homography solution is not so accurate as that from ESM tracking algorithm.

Therefore we combine the GPU-ESM tracking and GPU-SIFT methods to enhance the system performance. The combination model is shown in Fig.1. Both GPU-ESM tracking and GPU-SIFT run on GPUs simultaneously to process the input images. Though the two threads might process two different frames because of their different processing speed, the system can still work well because there is no large image difference between the two images in such a small delay time.

In GPU-ESM tracking algorithm, the ZNCC correlation value will be checked after processing each image to determine whether ESM tracking failure has happened or not. If tracking failure happens, GPU-ESM will automatically load current homography from GPU-SIFT and set it as the new initial value. By this means, the GPU-ESM tracking algorithm can continue working. Therefore, the whole homography-based visual servo system can work smoothly with high reliability at a high processing speed.

4 Experiments

Four experiments have been carried out to evaluate the system. The first two experiments are to evaluate the efficiency of our GPU-ESM tracking algorithm. The third is

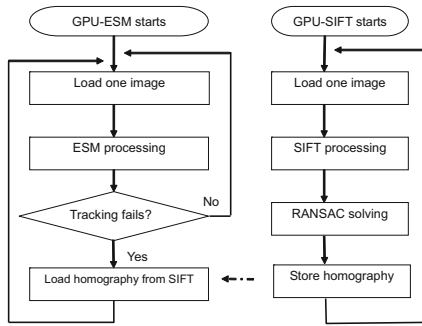


Fig. 1. Combination of GPU-ESM algorithm and GPU-SIFT

to verify the combination efficiency of both algorithms. 3D region tracking is developed in the last experiment. Images are captured from a 200 fps camera (Grasshopper GRAS-03K2M/C). Size is 640×480 .

4.1 Experiment I: Evaluation with Image Sequence

One image sequence (3000 frames of 640×480 grayscale images) are loaded into memory. Then the GPU-ESM and CPU-ESM process the same sequence from the memory. The number of ESM processing loop is set to 5. The tracking region size is chosen from 64×64 to 360×360 . Their processing speed (fps) and ratio are shown in Fig. 2

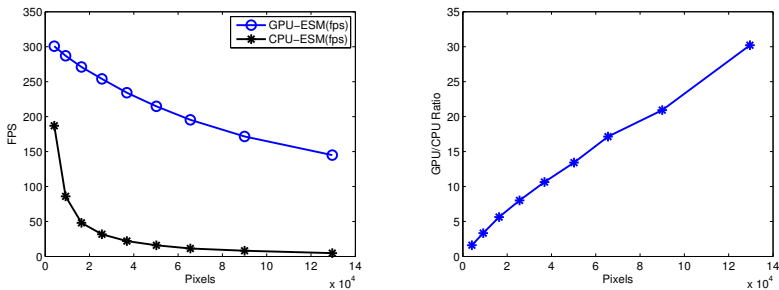


Fig. 2. Comparison on processing frame rate of GPU-ESM tracking and CPU-ESM tracking. X axis is the number of pixels in a square tracking region from 4096 (64×64) to 129600 (360×360).

Fig. 2 shows that GPU has greatly accelerated the ESM tracking algorithm. Though the processing speed of both decreases with the increase of tracking region size, GPU-ESM can still work at a relative high speed. As the ‘GPU/CPU Ratio’ increases with the pixel number, it also shows that GPU is more preferable for highly parallel tasks.

4.2 Experiment II: Evaluation with Real Application

Input images are from a 200 fps camera. The captured images are processed simultaneously by both GPU-ESM thread and CPU-ESM thread. Images extracted from

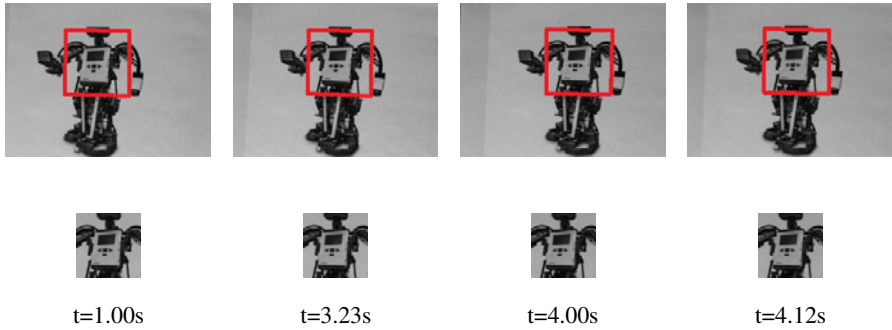


Fig. 3. GPU-ESM tracking. The second row shows the warped images from the boxed region in current images (the first row). The result that all warped images are nearly the same shows that GPU-ESM can track the fast moving object.

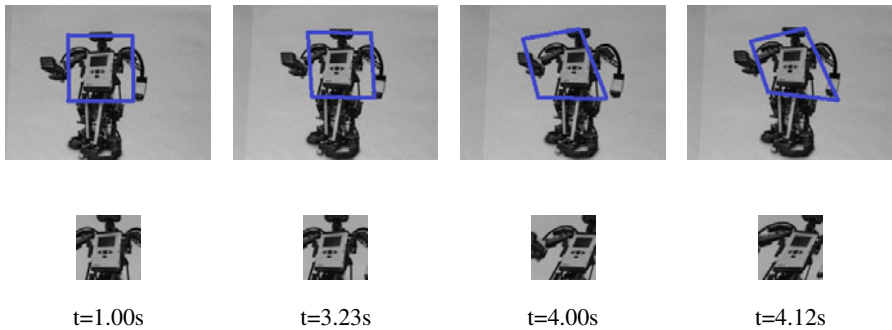


Fig. 4. CPU-ESM tracking. The change of warped images shows that CPU-ESM tracking can not track the same moving object as in Fig. 3.

the GPU-ESM and CPU-ESM tracking sequences are shown separately in Fig. 3 and Fig. 4. Tracking region is a 200×200 region shown in $t = 1.00s$.

The boxed regions in the first row of Fig. 3 and Fig. 4 are warped back with homography and shown in their second rows. Despite illumination change and image noise, the warped regions should be very close to the reference template when the tracking is accurately performed. During the experiment, we start to move the object from $t = 1.00s$. From the sequences in Fig. 4 we can see the CPU-ESM performs poorly with moving object (from $t = 3.23s$ tracking error happens, for $t > 4.00s$ the warped regions are totally different from the warped region of $t = 1.00s$) while GPU can still perform visual tracking well (the warped regions in Fig. 3 are nearly the same). This experiment shows that our system performance has been greatly enhanced by GPU acceleration.

4.3 Experiment III: Combination Evaluation

Occlusions are added to test the combination performance. The lower ZNCC threshold is set to 0.6 and the ZNCC value of each frame is plotted in Fig. 5. When occlusion

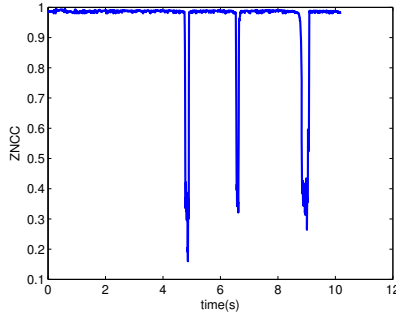


Fig. 5. ZNCC values respecting to time

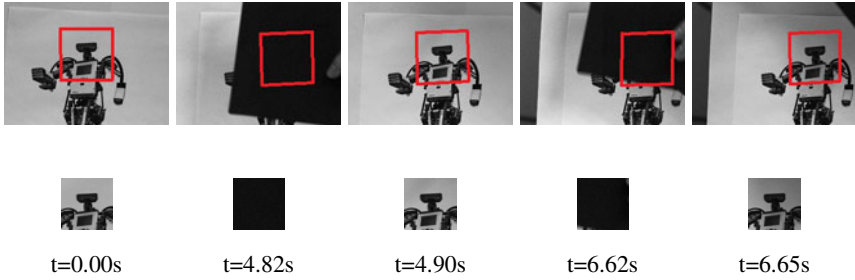


Fig. 6. Combination evaluation. Occlusion happens at $t = 4.82s$ and $t = 6.62s$. After occlusion is moved, it can continue tracking (see $t = 4.90s$, $t = 6.65s$). The red boxes shown in the first row show that by loading from SIFT, the ESM tracking can continue working even with occlusion.

happens at $t = 4.82s$ and $t = 6.62s$, the ZNCC value fell down (in Fig. 5). The GPU-ESM detected the tracking error and loaded the homography from GPU-SIFT. Therefore it can continue tracking at an acceptable accuracy. After occlusion is moved, the GPU-ESM can continue tracking (see the image sequences in Fig. 6). This has verified the effectiveness of our combination model.

4.4 Experiment IV: 3D Object Tracking

This experiment is to evaluate the 3D object tracking. Thanks to the GPU speedup, we extend the 2D planar tracking to 3D region tracking based on multiple planes tracking. In many applications a 3D tracking region can be separated into multiple adjacent planar regions. We can carry out ESM tracking on each planar region and merge the warped regions again to realize the tracking task. As shown in Fig. 7 for tracking area of 240×416 with two planar regions, the processing speed is 130 fps.

In our previous work of 3D region tracking, the boundaries of template region are manually chosen. Now the GPU-SIFT is also extend to 3D tracking. A 3D template region is chosen from current image (two adjacent regions in Fig. 7). Then the object is moved to a random initial pose(see the warped image of $t = 0.00s$).GPU-ESM will continue loading the homographies for two regions from SIFT and tracking the 3D region on the moving object until $t = 1.12s$, when the GPU-ESM has found an accurate

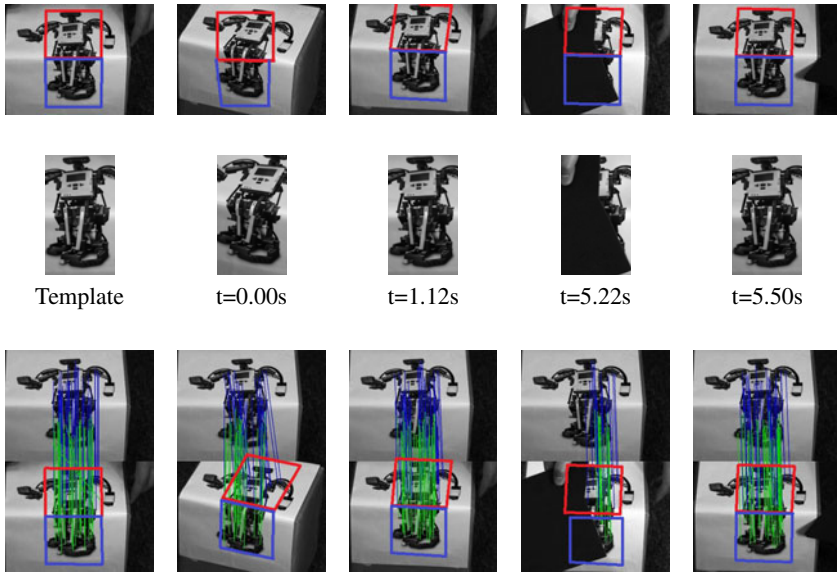


Fig. 7. 3D object tracking based on 2 planar regions tracking. Two adjacent tracking regions are described in red and blue boxes separately. The color boxes in first row show the tracked regions in current images. The warped images are shown in the second row. SIFT tracking results are shown in the third row. The tracked region by SIFT is also shown with color boxes. Occlusion happens at $t = 5.22s$ and disappears at $t = 5.50s$.

homography solution. The tracked regions at $t = 5.22s$ and $t = 5.50s$ show that our system can still track the 3D region by using SIFT results even when occlusion happens.

5 Optimization

Though CUDA uses C language with several extensions which makes it easier than other GPU languages, to make GPU code highly proficient, carefully optimization must be exploited and several important factors must be considered. In this section, we describe our optimization experience in our GPU applications.

5.1 Memory Hierarchy

CUDA provides a hierarchy of memory resources including on-chip memory (register, shared memory) and off-chip memory (local memory, global memory const and texture memory). In our GPU applications, we intensively utilize the fast on-chip shared memory instead of the long-latency global memory. For example, in kernel “Jesm”, the computation of J_{esm} matrix needs several intermediate results based on the image gradient. So we first load the image gradient data into shared memory and then continue other computation from shared memory. By using this “cache” like strategy, we have greatly reduced the kernel’s running time.

We use texture memory in kernel “Warping”. With the bilinear filter function by CUDA, we only need to set the filter mode parameter to bilinear. When fetching the texture memory, the returned value is computed automatically based on the input coordinates with the bilinear filter. This hardware function helps us to skip its programming.

5.2 Memory Coalescing

By using memory coalescing in CUDA, a half warp of 16 GPU threads can finish 16 global data fetching in as few as 1 or 2 transactions. In our applications, we have intensively used this optimization technique. For example, to calculate the mean \bar{I} in ZNCC correlation of a 360×360 region, first we need to calculate the sum of I . We use 1 block of 512 threads (the index of each thread “threadID” is from 0 to 511) to accumulate all the 129600 pixels. As $129600 = 254 * 510 + 60$, the number of data processed by each thread is 254 (except the last two threads with only 60 data). The normal idea is using “for-loop” in each GPU thread like this:

$$\text{for}(j = \text{threadID} * 254; j < (\text{threadID} + 1) * 254; j++) \{ \text{sum} += I[j]; \}$$

To fully use memory coalescing, we change the code to follows:

$$\text{for}(j = \text{threadID}; j < 129600; j += 512) \{ \text{sum} += I[j]; \}$$

Both “for-loops” seem to have same performance for a GPU thread. But due to GPU’s particular memory fetching mechanism, speedup really happens on GPU.

GPU memory is accessed in a continuous block mode, i.e. during one GPU memory access, data from a block of continuously addressing memory space will be loaded simultaneously. For example, it can load $T[0] \sim T[15]$ simultaneously by 16 GPU threads. In the latter loop, the fetched 16 data can be parallel processed by 16 GPU threads. Meanwhile in the former loop, only 1 data of these 16 data is used by 1 thread while all other 15 data is deserted. Each of other 15 threads must invoke other 15 GPU memory access to fetch their own data. Therefore, for the same data fetching, the former loop costs about 15 times more memory access time than the latter loop. With memory coalescing strategy shown in the latter loop, we have substantially reduced the total number of running time.

6 Conclusions

In this paper, an efficient combination approach of GPU-ESM and GPU-SIFT is presented. Experimental results verified the efficiency and effectiveness of our approach. The optimization techniques in our implementations are presented as a reference for other GPU application developers.

References

1. Szeliski, R.: Handbook of Mathematical Models in Computer Vision, pp. 273–292. Springer, Heidelberg (2006)

2. Baker, S., Matthews, I.: Lucas-Kanade 20 Years On: A Unifying Framework. *International Journal of Computer Vision* 56(3), 221–255 (2004)
3. Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*, pp. 11–23. Cambridge University Press, Cambridge (2000)
4. Lucas, B.D., Kanade, T.: An Iterative Image Registration Technique with an Application to Stereo Vision. In: *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 674–679 (1981)
5. Tomasi, C., Kanade, T.: *Detection and Tracking of Point Features.*, Carnegie Mellon University Technical Report CMU-CS-91-132 (April 1991)
6. Benhimane, S., Malis, E.: Real-time image-based tracking of planes using efficient second-order minimization. In: *IEEE/RSJ International Conference on Intelligent Robots Systems*, Sendai, Japan (2004)
7. Malis, E.: Improving vision-based control using efficient second-order minimization techniques. In: *IEEE International Conference on Robotics and Automation*, New Orleans, USA (April 2004)
8. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60(2), 91–110 (2004)
9. Zang, C., Hashimoto, K.: GPU acceleration in a Visual Servo System. In: *ICAM 2010*, Osaka, Japan (October 2010)
10. Malis, E.: An efficient unified approach to direct visual tracking of rigid and deformable surfaces. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, USA (October 2007)
11. Lowe, David, G.: Object recognition from local scale-invariant features. In: *Proceedings of the International Conference on Computer Vision*, vol. 2, pp. 1150–1157 (1999)
12. Morel, J.M., Yu, G.: ASIFT: A New Framework for Fully Affine Invariant Image Comparison. *SIAM Journal on Imaging Sciences* 2(2), 438–469 (2009)
13. Ke, Y., Sukthankar, R.: PCA-SIFT: A more distinctive representation for local image descriptors. In: *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 506–513 (2004)
14. Bay, H., Tuytelaars, Y., Van, G.L.: Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding* 110, 346–359 (2008)
15. Fung, J., Mann, S.: OpenVIDIA: parallel GPU computer vision. *ACM MULTIMEDIA*, 849–852 (2005)
16. Fung, J., Mann, S.: Computer Vision Signal Processing on Graphics Processing Units. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. V-93–V-96 (2004)
17. Sinha, S., Frahm, J.-M., Pollefeys, M., Genc, Y.: Feature tracking and matching in video using programmable graphics hardware. *Machine Vision and Applications* (March 2007)
18. Heymann, S., Muller, K., Smolic, A., Froehlich, B., Wiegand, T.: SIFT implementation and optimization for general-purpose GPU. In: *WSCG 2007*, pp. 317–322 (2007)
19. <http://www.cs.unc.edu/~ccwu/siftgpu/>

Recognizing Human Actions by Using Spatio-temporal Motion Descriptors

Ákos Utasi and Andrea Kovács

Hungarian Academy of Sciences, Computer and Automation Research Institute
Distributed Events Analysis Research Group
Kende u. 13-17. H-1111 Budapest, Hungary
{utasi, andrea.kovacs}@sztaki.hu

Abstract. This paper presents a novel tool for detecting human actions in stationary surveillance camera videos. In the proposed method there is no need to detect and track the human body or to detect the spatial or spatio-temporal interest points of the events. Instead our method computes single-scale spatio-temporal descriptors to characterize the action patterns. Two different descriptors are evaluated: histograms of optical flow directions and histograms of frame difference gradients. The integral video method is also presented to improve the performance of the extraction of these features. We evaluated our methods on two datasets: a public dataset containing actions of persons drinking and a new dataset containing stand up events. According to our experiments both detectors are suitable for indoor applications and provide a robust tool for practical problems such as moving background, or partial occlusion.

Keywords: Human action recognition, optical flow, frame difference.

1 Introduction

In the last decade human action detection and recognition in video streams have been an active field of research. They can often be a prerequisite for applications such as visual surveillance, semantic video annotation/indexing and retrieval, or higher level video analysis. It is still a challenging problem due to the variations in body size and shape, clothing, or the diverse characteristic (e.g. velocity, gait, posture) of the actions performed by different actors. The environmental noise (e.g. illumination change, shadows, occlusion, moving or cluttered background) also increases the complexity of the problem.

Several methods have been developed for detecting objects (e.g. human body, face, vehicle) in static images, and some of the concepts have been extended for recognizing action in video sequences. Most of these methods rely on the sparsely detected interest points and features extracted at the location of these points. Our approach is also inspired by object detection approaches, but contrary to other methods we neglect the interest points, instead we create a dense grid of local statistics in a predefined size spatio-temporal window containing the whole

action. This single-scale descriptor can be used to scan across the multi-scale representation of the input video segments.

The rest of the paper is organized as follows. In Sec. 2 we briefly present related work in human action recognition. Our method is introduced in Sec. 3, including the parameter settings we used in our experiments. The datasets we used for evaluating the proposed methods are discussed in Sec. 4. In Sec. 5 we give our experimental results. Finally Sec. 6 concludes the paper.

2 Related Work

Early approaches to human behavior recognition are based on the detection and tracking of the body. Spatial 2-D or 3-D features are extracted at each time step and the time series of these features provide the description of the action to be recognized. For a broad overview of these approaches see [1]. Instead of object tracking, several approaches track the spatial features and perform recognition on the feature tracks (e.g. the collection of body parts [2] or view-invariant representation of trajectories [3]).

Most of the recent methods first employ sparse spatio-temporal interest point (feature point) detection (e.g. [4,5,6]) and extract features at the location of these points. Finally, the extracted feature set is used to distinguish the different action classes. Dollár et al. [4] extend spatial interest point detection in the spatio-temporal case by applying temporal 1-D Gabor filters, and tuned the detector to evoke strong responses to periodic motions (other spatio-temporal corners also have strong responses). At each interest point normalized pixel values, brightness gradients and optical flow are extracted. The descriptors are computed by concatenating all the gradients in a region, the dimension is reduced by Principal Component Analysis (PCA). The space-time extension of the Harris operator is used by Schüldt et al. [7] to find spatio-temporal feature points, and local features are combined with a Support Vector Machine (SVM) [8] classifier to recognize the action. Kläser et al. in [9] use the same interest point detector and propose the HOG3D descriptor, which is based on histograms of 3-D image gradient orientations and combines shape and motion information. Laptev et al. in [10] extract the histogram of gradients and the histogram of optical flow features in the location of space-time interest points, and the extracted features are represented as bag-of-features.

Several existing approaches do not use interest point detection. Efros et al. in [11] introduce an optical flow based motion descriptor, and use global measurement for the whole stabilized and figure-centric sequence. In [12] the original histogram of gradient-based approach [13] was extended by motion information, which was achieved by using optical flow orientation of two consecutive frames. The detection is performed in a spatial window similarly to the original method. In [14] each event was represented by a spatio-temporal volume containing motion and shape features. Havasi et al. in [15] present a real-time tracking method to recognize one specific event, the human walk. The proposed descriptor is based on the structural changes of human legs, and achieved high detection rate in indoor and outdoor scenes using several different classifiers.

Our method is closely related to the work of [12] and [14]. The main difference is that instead of image gradients we calculate the gradients of the difference of consecutive video frames, therefore our descriptor contains motion information only. Moreover, we evaluate two special arrangements for the quantization of the directions. An action is represented by these features in a fixed size spatio-temporal rectangular cuboid, and this single 3-D descriptor (on single scales) can be used for recognizing actions on the multi-scale representation of the input video segments.

3 Proposed Method

The proposed method is based on the original HOG-based human detector of [13], however we extended it with a third dimension. That is we calculate our features in spatio-temporal rectangular cuboids. Two different motion features are extracted: histograms of optical flow directions (*HFD*) and histograms of frame difference gradients (*HDG*).

3.1 Optical Flow Directions

For each frame in the sequence we extracted the optical flow vectors using the implementation [16] of the method proposed by [17]. From the $f_i = (x_i, y_i, vx_i, vy_i)$ optical flow vector its direction d_i and magnitude m_i is computed using

$$d_i = \frac{360}{2\pi} \tan^{-1} \left(\frac{vy_i}{vx_i} \right) \tag{1}$$

$$m_i = \sqrt{vx_i^2 + vy_i^2}, \tag{2}$$

then the directions are linearly quantized. Fig. 1(a) and (b) present two possible arrangements. In our experiments we evaluated both types.

3.2 Frame Difference Gradients

Let $g_i = (gx_i, gy_i)$ denote the gradient of the absolute difference of two consecutive video frames at position i , calculated by using 3×3 vertical and horizontal Sobel operator. The orientation and the magnitude of g_i can be calculated by using Eq. 1. Finally, the orientations are quantized using an arrangement presented in Fig. 1.

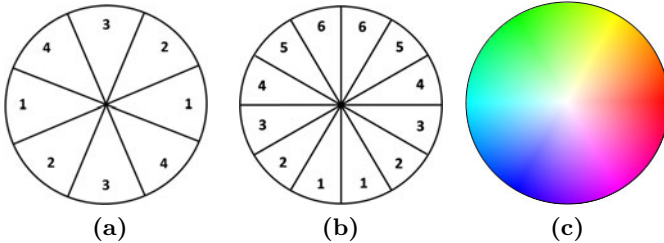


Fig. 1. (a) and (b) are two possible arrangements of quantization; (c) is the range of directions (hue) in the HSL color space and is used for visualizing our features

3.3 Descriptors

We use the original terms of [13] with the temporal extension. A *cell* is a small spatio-temporal rectangular cuboid of 8×8 pixels and $1/6$ sec duration, and in each *cell* a 4, 6 or 8-bin histogram is calculated from the quantized optical flow directions and frame difference gradient directions, while the magnitudes are used for weighted voting. A *block* is created as a group of several adjacent *cells* ($2 \times 2 \times 2$ *cells* in our experiments), and is used for normalizing the histograms of the *cells*. The features are the normalized histograms: *HFD* and *HDG*. A detection *window* of 96×128 pixels and 1 sec duration is tiled by these overlapping *blocks*. The features (normalized histograms of the *blocks*) in the spatio-temporal *window* are concatenated to form a vector and are used to recognize the event. Fig. 2 demonstrates the extracted optical flow *cell* histograms before *block* normalization, (a) and (b) are two samples taken from the positive dataset containing stand up events, while (c) is one sample from the negative set. Each direction in the 6-bin histogram is represented by 10 pixels in the *cell*, color is determined in the HSL color space according to

$$H_i = (i - 1/2) \times 30^\circ - 90^\circ \quad (3)$$

$$S_i = 1 \quad (4)$$

$$L_i = 0.5 \times h_i, \quad (5)$$

where h_i is the histogram value of the i th bin, and H_i is determined as the mean direction of the bin (see Fig. 1(b) and (c)). To express the direction of the motion the center pixels are represented by the bin with the highest h_i value.

Since each descriptor is single-scale, the detection *window* can be used to scan the multi-scale representation of the input video segments, and the extracted feature vectors are used in SVM classifier to recognize the action. In our experiments to obtain the desired 1 sec long *window* we applied temporal nearest neighbor

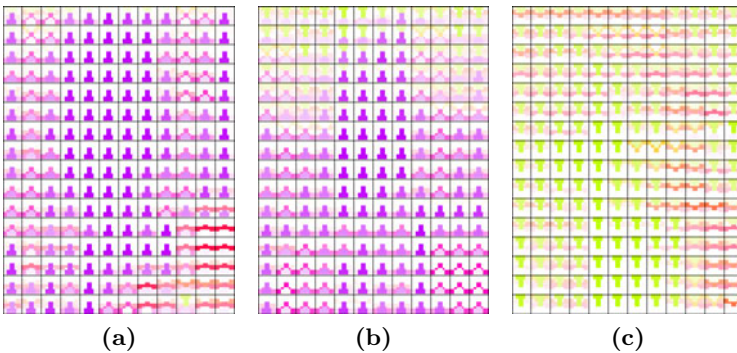


Fig. 2. Extracted histograms in the *cells* without *block* normalization, taken from the same temporal positions; (a) and (b) are from the positive sample set (stand up event); (c) is one sample from the negative set. The hue value in the HSL color space is used for visualizing the directions of the bins (see Fig. 1(c)).

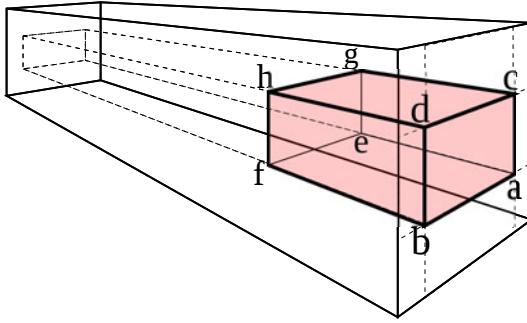


Fig. 3. Integral video representation: the sum in any rectangular cuboid can be computed using eight references

interpolation on the optical flow vectors and on the frame difference gradients extracted from the input video segments. In practice this means that in case of shorter segments the optical flow and the frame difference of some video frames were duplicated, while in case of larger duration several were dropped.

3.4 Integral Video Representation

The integral image representation proposed in [18] speeds up the computation of any rectangular sum, and therefore can be used to compute histograms efficiently. This technique can be extended to three dimensions. The integral video (*iv*) in 3-D is defined as follows. Let $f(x, y, t)$ denote the pixel at position (x, y) of the video frame at time t , then the *iv* is defined as

$$iv(x, y, t) = \sum_{x' \leq x, y' \leq y, t' \leq t} f(x', y', t'). \quad (6)$$

Extending the original recurrences to 3D we will get

$$sf(x, y, t) = sf(x, y, t - 1) + f(x, y, t) \quad (7)$$

$$sr(x, y, t) = sr(x, y - 1, t) + sf(x, y, t) \quad (8)$$

$$iv(x, y, t) = iv(x - 1, y, t) + sr(x, y, t), \quad (9)$$

where $sf(x, y, t)$ is the cumulative frame sum at position (x, y) , $sr(x, y, t)$ is the cumulative row sum at time t , $sf(x, y, -1) = 0$, $sr(x, -1, t) = 0$, and $iv(-1, y, t) = 0$. Using the integral group of frames any sum can be computed in rectangular cuboid (Fig. 3) using eight references as $a + d - (b + c) + f + g - (e + h)$.

4 Datasets

Our tests were performed on two datasets: one is publicly available and one was recorded in our offices.

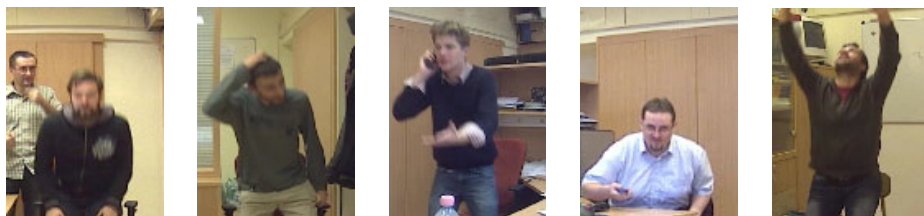


Fig. 4. Example video frames from the positive samples (frontal stand up events). Practical problems (e.g. motion in background) and inner-class variability (e.g. hand gestures, phone call) are clearly visible.

4.1 Drinking Dataset

To evaluate our method from the existing public datasets we used the drinking events introduced in [14]. However, on the author’s website only a limited number of shots are publicly available (33 shots in a single avi file). Therefore, our tests are also limited to these shots. Negative training and test samples were created by cropping random sized video parts from random temporal positions. The negative dataset contains 39 samples.

4.2 Stand Up Dataset¹

Most of existing datasets were recorded in controlled environment, hence we started to develop a new realistic dataset recorded in indoor environment. During the development we focused on practical problems such as moving objects in the background, occlusion or hand movements of the actors. Videos were recorded in our office by an ordinary IP camera. The dataset currently contains actions of six actors recorded at seven different scenes. For the recordings we used the camera’s own software, which used a standard MPEG-4 ASP coder at 1200 kbps rate for compression. The videos were recorded at 640×480 fsize and 30 fps rate.

Positive samples. From the recordings we manually cropped the frontal stand up events using a window with 0.75 aspect ratio. This window contained the body from the knee to the head with several extra pixels at the borders. Finally the windows were resized to 96×128 pixels using bicubic interpolation. In this set the duration of the events falls between 0.58 sec (18 frames) and 1.37 sec (41 frames). Currently the dataset contains 72 video sequences of the event. Fig. 4 presents example frames from the dataset.

Negative samples. We used two sets as negative samples. For the first set we manually selected some segments where different types of actions/movements were present and used the same method for resizing as we used for the positive samples. This dataset currently contains 67 video sequences. The second negative set was created by cropping random sized spatio-temporal windows (assuming

¹ Publicly available at http://web.eee.sztaki.hu/~ucu/sztaki_standup.tar.gz

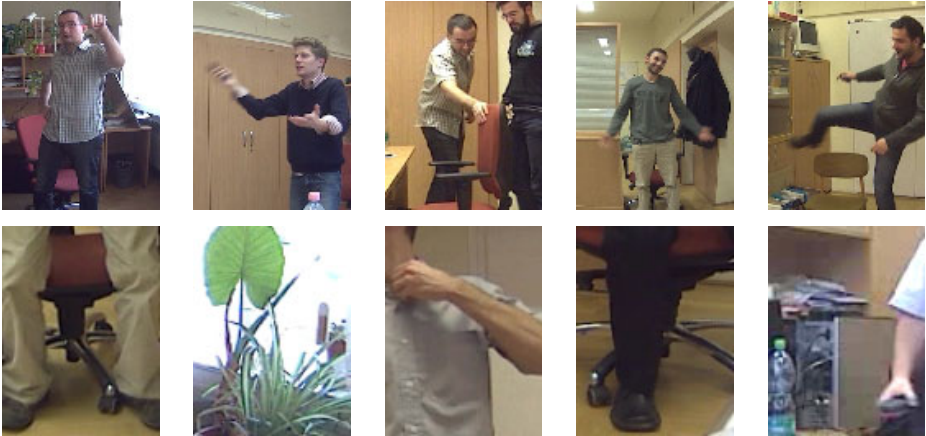


Fig. 5. Example video frames from the negative samples: hand selected video sequences (top); randomly cropped from the recordings (bottom)

0.75 aspect ratio) at random spatial and temporal positions. This dataset contains 63 sequences. The two negative sample sets currently contain 130 video sequences. Fig. 5 demonstrates some frames from the two negative samples.

5 Experimental Results

In our experiments we tested the two descriptors presented in Sec. 3.1 and Sec. 3.2, and both quantization arrangements (see Fig. 1) were evaluated with 4, 6, and 8 bins. Moreover, we also included in our tests the publicly available implementation of the *HOG3D* descriptor [9] using the default settings.

In our first experiment we used the *drinking* dataset presented in Sec. 4.1 to evaluate the performance of our methods. Two tests were performed with different size of training data. In the first test for training we used 10 samples from the positive and 11 from the negative dataset, while in case of the second test we increased the number of training samples to 17 (positive) and 20 (negative). The rest of the datasets were used for testing, and the trained SVM was used to recognize the action. Table 1 presents the confusion matrix of the recognition results of the first test with the smaller training set. Here we show the best results obtained by each descriptor, which were achieved in both cases by using the arrangement presented in Fig. 1b with 6 bins. Due to the larger training sets used in the second test the number of False Negatives decreased to FN=0, while the number of False Positives changed to FP=1 for *HDG* and *HFD*. In case of the *HOG3D* we obtained FN=0 and FP=1.

We used our *stand up* dataset (see Sec. 4.2) in the second experiment. Again, two tests were performed with training data of different size. In the first test we used 24 samples from the positive and 36 from the negative dataset for training, while in case of the second test we increased the number of training samples to 41 (positive) and 87 (negative). Table 2 summarizes the recognition results of

Table 1. Confusion matrix of the recognition results of the first experiment, where 10 positive and 11 negative samples were used for training the SVM. The remaining data (23 positive and 28 negative) were used for evaluation.

		HDG		HFD		HOG3D	
		Positive	Negative	Positive	Negative	Positive	Negative
Reference	Positive	TP=18	FN=5	TP=19	FN=4	TP=23	FN=0
	Negative	FP=0	TN=28	FP=3	TN=25	FP=1	TN=27

Table 2. Confusion matrix of the recognition results of the second experiment. In this test 24 positive and 36 negative samples were used for training the SVM. The remaining data (48 positive and 94 negative) were used for evaluation.

		HDG		HFD		HOG3D	
		Positive	Negative	Positive	Negative	Positive	Negative
Reference	Positive	TP=46	FN=2	TP=48	FN=0	TP=48	FN=0
	Negative	FP=1	TN=93	FP=3	TN=91	FP=2	TN=92

Table 3. Computational costs of the different steps in the recognition procedure

<i>HDG</i> extraction	21.18 msec
<i>HFD</i> extraction	48.81 msec
<i>HOG3D</i> extraction	34.30 msec
SVM-based recognition	0.79 msec

the first test, where the *HDG*-based recognition resulted in a 95.83% TPR (true positive rate), and a 1.06% FPR (false positive rate), while using the *HFD*-based detector a TPR=100% and a FPR=3.19% were achieved. Please note that only the best results are presented, which were obtained by using the arrangement of Fig. 1b with 6 and 8 bins for the *HFD* and the arrangement of Fig. 1a with 6 and 8 bins for the *HDG*. In the second test by using the increased training set we obtained FP=1 and FN=0 for the *HFD* method, FP=1 and FN=2 for the *HDG*, FP=3 and FN=0 for the *HOG3D*.

Finally, we also measured the duration of each step in the recognition procedure. The computation results are summarized in Table 3.

6 Conclusion

In this paper we presented a novel approach for recognizing human action. We used two different spatio-temporal motion-based descriptors and different quantization arrangements to characterize the event. Instead of representing the action as a set of features extracted at interest point locations, in our approach one single feature describes the whole action. To test our method we used a publicly

available dataset, but additionally we created a dataset of persons standing up, which contained several types of practical problems (e.g. motion in background or partial occlusion). According to our experiments the simple frame-difference based descriptor achieved recognition rates comparable to the optical flow-based approach, with significantly lower computational complexity. In the future we are planning to increase the size of our current dataset and also the number of different action types. Moreover, we will also evaluate how the different parameter settings (e.g. quantization or *cell* and *block* size) affect the recognition performance.

Acknowledgement

This work was partially supported by the Hungarian Scientific Research Fund under grant number 76159.

References

1. Gavrila, D.M.: The visual analysis of human movement: a survey. *Computer Vision and Image Understanding* 73(1), 82–98 (1999)
2. Song, Y., Goncalves, L., Perona, P.: Unsupervised learning of human motion. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 25(7), 814–827 (2003)
3. Rao, C., Shah, M.: View-invariance in action recognition. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 316–322 (2001)
4. Dollár, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: *Proc. of the 14th Int. Conf. on Computer Communications and Networks*, pp. 65–72 (2005)
5. Laptev, I.: On space-time interest points. *Int. J. of Computer Vision* 64(2–3), 107–123 (2005)
6. Ballan, L., Bertini, M., Del Bimbo, A., Seidenari, L., Serra, G.: Recognizing human actions by fusing spatio-temporal appearance and motion descriptors. In: *Proc. of the IEEE Int. Conf. on Image Processing* (2009)
7. Schüldt, C., Laptev, I., Caputo, B.: Recognizing human actions: A local SVM approach. In: *Proc. of the 17th Int. Conf. on Pattern Recognition*, pp. 32–36 (2004)
8. Vapnik, V.N.: *Statistical Learning Theory*. Wiley Interscience, Hoboken (1998)
9. Kläser, A., Marszałek, M., Schmid, C.: A spatio-temporal descriptor based on 3D-gradients. In: *Proc. of the British Machine Vision Conference*, pp. 995–1004 (2008)
10. Laptev, I., Marszałek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1–8 (2008)
11. Efros, A.A., Berg, A.C., Mori, G., Malik, J.: Recognizing action at a distance. In: *Proc. of the 9th IEEE Int. Conf. on Computer Vision*, pp. 726–733 (2003)
12. Dalal, N., Triggs, B., Schmid, C.: Human detection using oriented histograms of flow and appearance. In: *Proc. of the European Conf. on Computer Vision*, pp. 7–13 (2006)
13. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 886–893 (2005)

14. Laptev, I., Pérez, P.: Retrieving actions in movies. In: Proc. of the IEEE Int. Conf. on Computer Vision (2007)
15. Havasi, L., Szilávik, Z., Szirányi, T.: Higher order symmetry for non-linear classification of human walk detection. *Pattern Recognition Letters* 27(7), 822–829 (2006)
16. Galvin, B., McCane, B., Novins, K., Mason, D., Mills, S.: Recovering motion fields: An evaluation of eight optical flow algorithms. In: Proc. of the British Machine Vision Conference, pp. 195–204 (1998)
17. Proesmans, M., Van Gool, L.J., Pauwels, E.J., Oosterlinck, A.: Determination of optical flow and its discontinuities using non-linear diffusion. In: Proc. of the 3rd European Conf. on Computer Vision, pp. 295–304 (1994)
18. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, pp. 511–518 (2004)

Author Index

- Aelterman, Jan II-46
Aghajan, Hamid II-131
Akman, Oytun I-290
Alimi, Adel M. II-110
Angulo, Jesús I-426, I-452
Anwar, Adeel I-10
Aziz, Muhammad Zaheer I-367
- Balázs, Péter I-390
Bansal, Manu II-34
Barbieri, Mauro II-1
Barman, Sarah I-135, II-345
Barnich, Olivier II-98
Beghin, Thibaut II-345
Ben Amar, Chokri II-110
Ben Aoun, Najib II-110
Benezeth, Yannick II-121
Berry, François I-344
Bieber, Etienne II-203
Blumenstein, Michael I-145
Bonnier, Nicolas I-114
Borkar, Amol II-179
Boukrouche, Abdelhani I-105
- Cazorla, Miguel I-464
Chan, Yuk Hin I-332
Chen, Haibing II-69
Chen, Ting-Wen I-321
Chetty, Girija II-79
Chi, Zheru I-62, I-72
Chmelar, Petr II-155
Cope, James S. I-135, II-345
Coppens, Jonas I-221
Cornelis, Jan I-267
Corrales-Garcia, Alberto II-22
Cyganek, Bogusław II-191
- Deboeverie, Francis I-173
Delmas, Patrice I-332, I-476
De Meulemeester, Simon I-309
Denis, Leon I-267
Despotović, Ivana I-153
De Vylder, Jonas I-125
de With, Peter H.N. II-1
Díaz-Chito, Katerine II-304
- Díaz-Villanueva, Wladimiro II-304
Djouani, Karim II-227
D’Orazio, Tiziana II-143
Dornaika, Fadi II-239
Douterloigne, Koen I-125, II-13
- Emile, Bruno II-121
- Fabián, Tomáš I-402
Feng, Dagan I-62, I-72
Fernandez-Escribano, Gerardo II-22
Ferri, Francesc J. II-304
Fischmeister, Edgar I-50
Francos, Joseph M. I-93
Fu, Hong I-62, I-72
- Garcia Alvarez, Julio Cesar I-309
Gaura, Jan I-402
Gautama, Sidharta II-13
Gholamhosseini, Hamid I-300
Gimel’farb, Georgy I-332, I-476
Gong, Weiguo II-284
Goodwin, Julian II-79
Goossens, Bart II-46
Gu, Xiaohua II-284
- Hafiane, Adel I-105
Hamam, Yskandar II-227
Hammoudi, Karim II-239
Hashimoto, Koichi II-354
Hayes, Monson II-179
He, Xiangjian I-233
Hemery, Baptiste II-121
Herout, Adam II-215
Heyvaert, Michaël I-221
Horé, Alain I-197
Houam, Lotfi I-105
Hradiš, Michal II-215
Huang, Xiaohua II-312
Huber-Mörk, Reinhold I-50
- Ichikawa, Makoto I-185
Isaza, Cesar I-30

- Jelača, Vedran I-153
 Jennane, Rachid I-105
 Jia, Wenjing I-233
 Jiang, Gangyi II-69
 Jonker, Pieter I-290
 Jubran, Mohammad II-34

 Kaarna, Arto I-80, II-261
 Kang, Hyun-Soo I-300
 Karray, Hichem II-110
 Kim, Soo K. II-249
 Kim, Tae-Jung II-58
 Klepaczko, Artur I-245
 Kölsch, Mathias II-292
 Kondi, Lisimachos P. II-34
 Kovács, Andrea I-163, II-272, II-366
 Krim, Hamid I-279
 Krumnikl, Michal I-402

 Lam, Kin-Man I-233
 Lanik, Ales II-155
 Laurent, Hélène II-121
 Lee, Deokwoo I-279
 Lee, Kang-San I-300
 Leo, Marco II-143
 Lespessailles, Eric I-105
 Li, Fucui II-69
 Li, Jianmin I-233
 Li, Weihong II-284
 Liang, Zhen I-62
 Lin, Huei-Yung I-321
 Liu, Yuehu I-357
 Lukin, Vladimir II-261
 Luong, Hiêp II-46

 Ma, Zhonghua II-167
 Magnier, Baptiste I-209
 Mansoor, Atif Bin I-10
 Mazzeo, Pier Luigi II-143
 Meedeniya, Adrian I-145
 Mertsching, Bärbel I-367
 Mian, Ajmal II-332
 Miike, Hidetoshi I-185
 Mlich, Jozef II-155
 Monnin, David II-203
 Montesinos, Philippe I-209
 Munteanu, Adrian I-267

 Nagy, Antal I-390
 Ngan, King Ngi I-255

 Noel, Guillaume II-227
 Nölle, Michael I-50
 Nomura, Atsushi I-185

 Oberhauser, Andreas I-50
 Okada, Koichi I-185
 Orjuela Vargas, Sergio Alejandro I-309
 Ortiz Jaramillo, Benhur I-309
 Othman, Ahmed A. I-38

 Pahjehfouladgaran, Maryam I-80
 Paradowski, Mariusz I-18
 Pelissier, Frantz I-344
 Peng, Zongju II-69
 Pham, Tuan Q. I-438
 Philips, Wilfried I-125, I-153, I-173,
 I-309, II-13, II-46, II-88
 Piérard, Sébastien II-98
 Pietikäinen, Matti II-312
 Pizurica, Aleksandra I-309, II-46
 Ponomarenko, Nikolay II-261

 Quiles, Francisco Jose II-22

 Raducanu, Bogdan I-30
 Ranganath, Heggere S. II-249
 Remagnino, Paolo I-135, II-345
 Riddle, Patricia I-476
 Romero, Anna I-464
 Rooms, Filip I-309
 Rosenberger, Christophe II-121

 Sai, Ilya S. I-1
 Sai, Sergey V. I-1
 Salas, Joaquin I-30
 Satti, Shahid M. I-267
 Schelkens, Peter I-267
 Schmitt, Gwenaël II-203
 Schneider, Armin II-203
 Scott, Adele F. II-167
 Seo, Bo-Seok II-58
 Shadkami, Pasha I-114
 Shan, Caifeng II-323
 Shao, Feng II-69
 Sherrah, Jamie I-414
 Shorin, Al I-476
 Shrestha, Prarthana II-1
 Singh, Monica II-79
 Śluzek, Andrzej I-18
 Smith, Mark T. II-179
 Sojka, Eduard I-402

- Sorokin, Nikolay Yu. I-1
Spagnolo, Paolo II-143
Su, Yuanqi I-357
Suh, Jae-Won II-58
Szczypiński, Piotr I-245
Sziranyi, Tamas I-163, II-272
- Teelen, Kristof I-173, II-88
Tizhoosh, Hamid R. I-38
Tuxworth, Gervase I-145
- Utasi, Ákos II-366
- Valkenburg, Robert I-332
Van Droogenbroeck, Marc II-98
Van Hamme, David I-221, II-88
Vansteenkiste, Ewout I-153
Varga, László I-390
Veelaert, Peter I-173, I-221, II-88
Velasco-Forero, Santiago I-452
Vendrig, Jeroen II-167
Vigdor, Boaz I-93
- Wali, Ali II-110
Wang, Lin I-233
Weda, Hans II-1
Wedge, Daniel II-167
Wei, Daming I-233
Wilkin, Paul I-135
Wu, Chen II-131
Wu, Qiang I-233
- Yang, Liping II-284
Yang, Yang I-357
Yu, Mei II-69
- Zang, Chuantao II-354
Zemčík, Pavel II-215
Zenou, Emmanuel I-379
Zhang, Qian I-255
Zhang, Siyuan I-379
Zhao, Guoying II-312
Zheng, Wenming II-312
Ziou, Djemel I-197
Zriakhov, Mikhail II-261