

Ron Kimmel
Reinhard Klette
Akihiro Sugimoto (Eds.)

LNCS 6495

Computer Vision – ACCV 2010

10th Asian Conference on Computer Vision
Queenstown, New Zealand, November 2010
Revised Selected Papers, Part IV

4 Part IV



 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

Ron Kimmel Reinhard Klette
Akihiro Sugimoto (Eds.)

Computer Vision – ACCV 2010

10th Asian Conference on Computer Vision
Queenstown, New Zealand, November 8-12, 2010
Revised Selected Papers, Part IV

Volume Editors

Ron Kimmel
Department of Computer Science
Technion – Israel Institute of Technology
Haifa 32000, Israel
E-mail: ron@cs.technion.ac.il

Reinhard Klette
The University of Auckland
Private Bag 92019, Auckland 1142, New Zealand
E-mail: r.klette@auckland.ac.nz

Akihiro Sugimoto
National Institute of Informatics
Chiyoda, Tokyo 1018430, Japan
E-mail: sugimoto@nii.ac.jp

ISSN 0302-9743
ISBN 978-3-642-19281-4
DOI 10.1007/978-3-642-19282-1

e-ISSN 1611-3349
e-ISBN 978-3-642-19282-1

Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011921594

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

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

© Springer-Verlag Berlin Heidelberg 2011

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.

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

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Coverpicture: Lake Wakatipu and the The Remarkables, from 'Skyline Queenstown' where the conference dinner took place.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

The 2010 Asian Conference on Computer Vision took place in the southern hemisphere, in “The Land of the Long White Cloud” in Maori language, also known as New Zealand, in the beautiful town of Queenstown. If we try to segment the world we realize that New Zealand does not belong officially to any continent. Similarly, in computer vision we often try to define outliers while attempting to segment images, separate them to well-defined “continents” we refer to as objects. Thus, the ACCV Steering Committee consciously chose this remote and pretty island as a perfect location for ACCV2010, to host the computer vision conference of the most populated and largest continent, Asia. Here, on South Island we studied and exchanged ideas about the most recent advances in image understanding and processing sciences.

Scientists from all well-defined continents (as well as ill-defined ones) submitted high-quality papers on subjects ranging from algorithms that attempt to automatically understand the content of images, optical methods coupled with computational techniques that enhance and improve images, and capturing and analyzing the world’s geometry while preparing for higher-level image and shape understanding. Novel geometry techniques, statistical-learning methods, and modern algebraic procedures rapidly propagate their way into this fascinating field as we witness in many of the papers one can find in this collection.

For this 2010 issue of ACCV, we had to select a relatively small part of all the submissions and did our best to solve the impossible ranking problem in the process. We had three keynote speakers (Sing Bing Kang lecturing on modeling of plants and trees, Sebastian Sylwan talking about computer vision in production of visual effects, and Tim Cootes lecturing about modelling deformable object), eight workshops (Computational Photography and Esthetics, Computer Vision in Vehicle Technology, e-Heritage, Gaze Sensing and Interactions, Subspace, Video Event Categorization, Tagging and Retrieval, Visual Surveillance, and Application of Computer Vision for Mixed and Augmented Reality), and four tutorials. Three Program Chairs and 38 Area Chairs finalized the decision about the selection of 35 oral presentations and 171 posters that were voted for out of 739, so far the highest number of ACCV, submissions. During the reviewing process we made sure that each paper was reviewed by at least three reviewers, we added a rebuttal phase for the first time in ACCV, and held a three-day AC meeting in Tokyo to finalize the non-trivial acceptance decision-making process.

Our sponsors were the Asian Federation of Computer Vision Societies (AFCV), NextWindow–Touch-Screen Technology, NICTA–Australia’s Information and Communications Technology (ICT), Microsoft Research Asia, Areograph–Interactive Computer Graphics, Adept Electronic Solutions, and 4D View Solutions.

Finally, the *International Journal of Computer Vision* (IJCV) sponsored the Best Student Paper Award.

We wish to acknowledge a number of people for their invaluable help in putting this conference together. Many thanks to the Organizing Committee for their excellent logistical management, the Area Chairs for their rigorous evaluation of papers, the Program Committee members as well as external reviewers for their considerable time and effort, and the authors for their outstanding contributions.

We also wish to acknowledge the following individuals for their tremendous service: Yoshihiko Mochizuki for support in Tokyo (especially also for the Area Chair meeting), Gisela Klette, Konstantin Schauwecker, and Simon Hermann for processing the 200+ Latex submissions for these proceedings, Kaye Saunders for running the conference office at Otago University, and the volunteer students during the conference from Otago University and the *.enpeda..* group at The University of Auckland. We also thank all the colleagues listed on the following pages who contributed to this conference in their specified roles, led by Brendan McCane who took the main responsibilities.

ACCV2010 was a very enjoyable conference. We hope that the next ACCV meetings will attract even more high-quality submissions.

November 2010

Ron Kimmel
Reinhard Klette
Akihiro Sugimoto



Organization

Steering Committee

Katsushi Ikeuchi	University of Tokyo, Japan
Tieniu Tan	Institute of Automation, Chinese Academy of Science, China
Chil-Woo Lee	Chonnam National University, Korea
Yasushi Yagi	Osaka University, Japan

Honorary Chairs

P. Anandan	Microsoft Research India
Richard Hartley	Australian National University, NICTA

General Chairs

Brendan McCane	University of Otago, New Zealand
Hongbin Zha	Peking University, China

Program Chairs

Ron Kimmel	Israel Institute of Technology
Reinhard Klette	University of Auckland, New Zealand
Akihiro Sugimoto	National Institute of Informatics, Japan

Local Organization Chairs

Brendan McCane	University of Otago, New Zealand
John Morris	University of Auckland, New Zealand

Workshop Chairs

Fay Huang	Ilan University, Yi-Lan, Taiwan
Reinhard Koch	University of Kiel, Germany

Tutorial Chair

Terrence Sim	National University of Singapore
--------------	----------------------------------

Demo Chairs

Kenji Irie	Lincoln Ventures, New Zealand
Alan McKinnon	Lincoln University, New Zealand

Publication Chairs

Michael Cree	University of Waikato, New Zealand
Keith Unsworth	Lincoln University, New Zealand

Publicity Chairs

John Barron	University of Western Ontario, Canada
Domingo Mery	Pontificia Universidad Católica de Chile
Ioannis Pitas	Aristotle University of Thessaloniki, Greece

Area Chairs

Donald G. Bailey	Massey University, Palmerston North, New Zealand
Horst Bischof	TU Graz, Austria
Alex Bronstein	Technion, Haifa, Israel
Michael S. Brown	National University of Singapore
Chu-Song Chen	Academia Sinica, Taipei, Taiwan
Hui Chen	Shandong University, Jinan, China
Laurent Cohen	University Paris Dauphine, France
Daniel Cremers	Bonn University, Germany
Eduardo Destefanis	Technical University Cordoba, Argentina
Hamid Krim	North Carolina State University, Raleigh, USA
Chil-Woo Lee	Chonnam National University, Gwangju, Korea
Facundo Memoli	Stanford University, USA
Kyoung Mu Lee	Seoul National University, Korea
Stephen Lin	Microsoft Research Asia, Beijing, China
Kai-Kuang Ma	Nanyang Technological University, Singapore
Niloy J. Mitra	Indian Institute of Technology, New Delhi, India
P.J. Narayanan	International Institute of Information Technology, Hyderabad, India
Nassir Navab	TU Munich, Germany
Takayuki Okatani	Tohoku University, Sendai City, Japan
Tomas Pajdla	Czech Technical University, Prague, Czech Republic
Nikos Paragios	Ecole Centrale de Paris, France
Robert Pless	Washington University, St. Louis, USA
Marc Pollefeys	ETH Zürich, Switzerland
Mariano Rivera	CIMAT Guanajuato, Mexico
Antonio Robles-Kelly	National ICT, Canberra, Australia
Hideo Saito	Keio University, Yokohama, Japan

Yoichi Sato	The University of Tokyo, Japan
Nicu Sebe	University of Trento, Italy
Stefano Soatto	University of California, Los Angeles, USA
Nir Sochen	Tel Aviv University, Israel
Peter Sturm	INRIA Grenoble, France
David Suter	University of Adelaide, Australia
Robby T. Tan	University of Utrecht, The Netherlands
Toshikazu Wada	Wakayama University, Japan
Yaser Yacoob	University of Maryland, College Park, USA
Ming-Hsuan Yang	University of California, Merced, USA
Hong Zhang	University of Alberta, Edmonton, Canada
Mengjie Zhang	Victoria University of Wellington, New Zealand

Program Committee Members

Abdenour, Hadid	Benosman, Ryad
Achard, Catherine	Berkels, Benjamin
Ai, Haizhou	Berthier, Michel
Aiger, Dror	Bhattacharya, Bhargab
Alahari, Karteek	Biswas, Prabir
Araguas, Gaston	Bo, Liefeng
Arica, Nafiz	Boerdgen, Markus
Ariki, Yasuo	Bors, Adrian
Arslan, Abdullah	Boshra, Michael
Astroem, Kalle	Bouguila, Nizar
August, Jonas	Boyer, Edmond
Aura Vese, Luminita	Bronstein, Michael
Azevedo-Marques, Paulo	Bruhn, Andres
Bagdanov, Andy	Buckley, Michael
Bagon, Shai	Cai, Jinhai
Bai, Xiang	Cai, Zhenjiang
Baloch, Sajjad	Calderón, Jesús
Baltes, Jacky	Camastra, Francesco
Bao, Yufang	Canavesio, Luisa
Bar, Leah	Cao, Xun
Barbu, Adrian	Carlo, Colombo
Barnes, Nick	Carlsson, Stefan
Barron, John	Caspi, Yaron
Bartoli, Adrien	Castellani, Umberto
Baust, Maximilian	Celik, Turgay
Ben Hamza, Abdessamad	Cham, Tat-Jen
BenAbdelkader, Chiraz	Chan, Antoni
Ben-ari, Rami	Chandran, Sharat
Beng-Jin, AndrewTeoh	Charvillat, Vincent

Chellappa, Rama
Chen, Bing-Yu
Chen, Chia-Yen
Chen, Chi-Fa
Chen, Haifeng
Chen, Hwann-Tzong
Chen, Jie
Chen, Jiun-Hung
Chen, Ling
Chen, Xiaowu
Chen, Xilin
Chen, Yong-Sheng
Cheng, Shyi-Chyi
Chia, Liang-Tien
Chien, Shao-Yi
Chin, Tat-Jun
Chuang, Yung-Yu
Chung, Albert
Chunhong, Pan
Civera, Javier
Coleman, Sonya
Cootes, Tim
Costeira, JoaoPaulo
Cristani, Marco
Csaba, Beleznai
Cui, Jinshi
Daniilidis, Kostas
Daras, Petros
Davis, Larry
De Campos, Teofilo
Demirci, Fatih
Deng, D. Jeremiah
Deng, Hongli
Denzler, Joachim
Derrode, Stephane
Diana, Mateus
Didas, Stephan
Dong, Qiulei
Donoser, Michael
Doretto, Gianfranco
Dorst, Leo
Duan, Fuqing
Dueck, Delbert
Duric, Zoran
Dutta Roy, Sumantra

Ebner, Marc
Einhauser, Wolfgang
Engels, Christopher
Eroglu-Erdem, Cigdem
Escolano, Francisco
Esteves, Claudia
Evans, Adrian
Fang, Wen-Pinn
Feigin, Micha
Feng, Jianjiang
Ferri, Francesc
Fite Georgel, Pierre
Flitti, Farid
Frahm, Jan-Michael
Francisco Giro Martín, Juan
Fraundorfer, Friedrich
Frosini, Patrizio
Fu, Chi-Wing
Fuh, Chiou-Shann
Fujiyoshi, Hironobu
Fukui, Kazuhiro
Fumera, Giorgio
Furst, Jacob
Fusiello, Andrea
Gall, Juergen
Gallup, David
Gang, Li
Gasparini, Simone
Geiger, Andreas
Gertych, Arkadiusz
Gevers, Theo
Glocker, Ben
Godin, Guy
Goecke, Roland
Goldluecke, Bastian
Goras, Bogdan
Gross, Ralph
Gu, I
Guerrero, Josechu
Guest, Richard
Guo, Guodong
Gupta, Abhinav
Gur, Yaniv
Hajebi, Kiana
Hall, Peter

Hamsici, Onur
Han, Bohyung
Hanbury, Allan
Harit, Gaurav
Hartley, Richard
HassabElgawi, Osman
Havlena, Michal
Hayes, Michael
Hayet, Jean-Bernard
He, Junfeng
Hee Han, Joon
Hiura, Shinsaku
Ho, Jeffrey
Ho, Yo-Sung
Ho Seo, Yung
Hollitt, Christopher
Hong, Hyunki
Hotta, Kazuhiro
Hotta, Seiji
Hou, Zujun
Hsu, Pai-Hui
Hua, Gang
Hua, Xian-Sheng
Huang, Chun-Rong
Huang, Fay
Huang, Kaiqi
Huang, Peter
Huang, Xiangsheng
Huang, Xiaolei
Hudelot, Celine
Hugo Sauchelli, Víctor
Hung, Yi-Ping
Hussein, Mohamed
Huynh, Cong Phuoc
Hyung Kim, Soo
Ichimura, Naoyuki
Ik Cho, Nam
Ikizler-Cinbis, Nazli
Il Park, Jong
Ilic, Slobodan
Imiya, Atsushi
Ishikawa, Hiroshi
Ishiyama, Rui
Iwai, Yoshio
Iwashita, Yumi
Jacobs, Nathan
Jafari-Khouzani, Kourosh
Jain, Arpit
Jannin, Pierre
Jawahar, C.V.
Jenkin, Michael
Jia, Jiaya
Jia, JinYuan
Jia, Yunde
Jiang, Shuqiang
Jiang, Xiaoyi
Jin Chung, Myung
Jo, Kang-Hyun
Johnson, Taylor
Joshi, Manjunath
Jurie, Frederic
Kagami, Shingo
Kakadiaris, Ioannis
Kale, Amit
Kamberov, George
Kanatani, Kenichi
Kankanhalli, Mohan
Kato, Zoltan
Katti, Harish
Kawakami, Rei
Kawasaki, Hiroshi
Keun Lee, Sang
Khan, Saad-Masood
Kim, Hansung
Kim, Kyungnam
Kim, Seon Joo
Kim, TaeHoon
Kita, Yasuyo
Kitahara, Itaru
Koepfler, Georges
Koeppen, Mario
Koeser, Kevin
Kokiopoulou, Effrosyni
Kokkinos, Iasonas
Kolesnikov, Alexander
Koschan, Andreas
Kotsiantis, Sotiris
Kown, Junghyun
Kruger, Norbert
Kuijper, Arjan

Kukenys, Ignas
 Kuno, Yoshinori
 Kuthirummal, Sujit
 Kwolek, Bogdan
 Kwon, Junseok
 Kybic, Jan
 Kyu Park, In
 Ladikos, Alexander
 Lai, Po-Hsiang
 Lai, Shang-Hong
 Lane, Richard
 Langs, Georg
 Lao, Shihong
 Lao, Zhiqiang
 Lauze, Francois
 Le, Duy-Dinh
 Le, Triet
 Lee, Jae-Ho
 Lee, Soochahn
 Leistner, Christian
 Leonardo, Bocchi
 Leow, Wee-Kheng
 Lepri, Bruno
 Lerasle, Frederic
 Li, Chunming
 Li, Hao
 Li, Hongdong
 Li, Stan
 Li, Yongmin
 Liao, T.Warren
 Lie, Wen-Nung
 Lien, Jenn-Jier
 Lim, Jongwoo
 Lim, Joo-Hwee
 Lin, Huei-Yung
 Lin, Weisi
 Lin, Wen-Chieh(Steve)
 Ling, Haibin
 Lipman, Yaron
 Liu, Cheng-Lin
 Liu, Jingen
 Liu, Ligang
 Liu, Qingshan
 Liu, Qingzhong
 Liu, Tianming

Liu, Tyng-Luh
 Liu, Xiaoming
 Liu, Yuncai
 Loog, Marco
 Lu, Huchuan
 Lu, Juwei
 Lu, Le
 Lucey, Simon
 Luo, Jiebo
 Macaire, Ludovic
 Maccormick, John
 Madabhushi, Anant
 Makris, Dimitrios
 Manabe, Yoshitsugu
 Marsland, Stephen
 Martinec, Daniel
 Martinet, Jean
 Martinez, Aleix
 Masuda, Takeshi
 Matsushita, Yasuyuki
 Mauthner, Thomas
 Maybank, Stephen
 McHenry, Kenton
 McNeill, Stephen
 Medioni, Gerard
 Mery, Domingo
 Mio, Washington
 Mittal, Anurag
 Miyazaki, Daisuke
 Mobahi, Hossein
 Moeslund, Thomas
 Mordohai, Philippos
 Moreno, Francesc
 Mori, Greg
 Mori, Kensaku
 Morris, John
 Mueller, Henning
 Mukaigawa, Yasuhiro
 Mukhopadhyay, Jayanta
 Muse, Pablo
 Nagahara, Hajime
 Nakajima, Shin-ichi
 Nanni, Loris
 Neshatian, Kouros
 Newsam, Shawn

Niethammer, Marc
 Nieuwenhuis, Claudia
 Nikos, Komodakis
 Nobuhara, Shohei
 Norimichi, Ukita
 Nozick, Vincent
 Ofek, Eyal
 Ohnishi, Naoya
 Oishi, Takeshi
 Okabe, Takahiro
 Okuma, Kenji
 Olague, Gustavo
 Omachi, Shinichiro
 Ovsjanikov, Maks
 Pankanti, Sharath
 Paquet, Thierry
 Paternak, Ofer
 Patras, Ioannis
 Pauly, Olivier
 Pavlovic, Vladimir
 Peers, Pieter
 Peng, Yigang
 Penman, David
 Pernici, Federico
 Petrou, Maria
 Ping, Wong Ya
 Prasad Mukherjee, Dipti
 Prati, Andrea
 Qian, Zhen
 Qin, Xueyin
 Raducanu, Bogdan
 Rafael Canali, Luis
 Rajashekar, Umesh
 Ramalingam, Srikumar
 Ray, Nilanjan
 Real, Pedro
 Remondino, Fabio
 Reulke, Ralf
 Reyes, EdelGarcia
 Ribeiro, Eraldo
 Riklin Raviv, Tammy
 Roberto, Tron
 Rosenhahn, Bodo
 Rosman, Guy
 Roth, Peter

Roy Chowdhury, Amit
 Rugis, John
 Ruiz Shulcloper, Jose
 Ruiz-Correa, Salvador
 Rusinkiewicz, Szymon
 Rustamov, Raif
 Sadri, Javad
 Saffari, Amir
 Saga, Satoshi
 Sagawa, Ryusuke
 Salzmann, Mathieu
 Sanchez, Jorge
 Sang, Nong
 Sang Hong, Ki
 Sang Lee, Guee
 Sappa, Angel
 Sarkis, Michel
 Sato, Imari
 Sato, Jun
 Sato, Tomokazu
 Schiele, Bernt
 Schikora, Marek
 Schoenemann, Thomas
 Scotney, Bryan
 Shan, Shiguang
 Sheikh, Yaser
 Shen, Chunhua
 Shi, Qinfeng
 Shih, Sheng-Wen
 Shimizu, Ikuko
 Shimshoni, Ilan
 Shin Park, You
 Sigal, Leonid
 Sinha, Sudeepa
 So Kweon, In
 Sommerlade, Eric
 Song, Andy
 Souvenir, Richard
 Srivastava, Anuj
 Staiano, Jacopo
 Stein, Gideon
 Stottinge, Julian
 Strecha, Christoph
 Strelakovski, Evgeny
 Subramanian, Ramanathan

Sugaya, Noriyuki
 Sumi, Yasushi
 Sun, Weidong
 Swaminathan, Rahul
 Tai, Yu-Wing
 Takamatsu, Jun
 Talbot, Hugues
 Tamaki, Toru
 Tan, Ping
 Tanaka, Masayuki
 Tang, Chi-Keung
 Tang, Jinshan
 Tang, Ming
 Taniguchi, Rinichiro
 Tao, Dacheng
 Tavares, João Manuel R.S.
 Teboul, Olivier
 Terauchi, Mutsuhiro
 Tian, Jing
 Tian, Taipeng
 Tobias, Reichl
 Toews, Matt
 Tominaga, Shoji
 Torii, Akihiko
 Tsin, Yanghai
 Turaga, Pavan
 Uchida, Seiichi
 Ueshiba, Toshio
 Unger, Markus
 Urtasun, Raquel
 van de Weijer, Joost
 Van Horebeek, Johan
 Vassallo, Raquel
 Vasseur, Pascal
 Vaswani, Namrata
 Wachinger, Christian
 Wang, Chen
 Wang, Cheng
 Wang, Hongcheng
 Wang, Jue
 Wang, Yu-Chiang
 Wang, Yunhong
 Wang, Zhi-Heng

Wang, Zhijie
 Wolf, Christian
 Wolf, Lior
 Wong, Kwan-Yee
 Woo, Young
 Wook Lee, Byung
 Wu, Jianxin
 Xue, Jianru
 Yagi, Yasushi
 Yan, Pingkun
 Yan, Shuicheng
 Yanai, Keiji
 Yang, Herbert
 Yang, Jie
 Yang, Yongliang
 Yi, June-Ho
 Yilmaz, Alper
 You, Suyi
 Yu, Jin
 Yu, Tianli
 Yuan, Junsong
 Yun, Il Dong
 Zach, Christopher
 Zelek, John
 Zha, Zheng-Jun
 Zhang, Cha
 Zhang, Changshui
 Zhang, Guofeng
 Zhang, Hongbin
 Zhang, Li
 Zhang, Liqing
 Zhang, Xiaoqin
 Zheng, Lu
 Zheng, Wenming
 Zhong, Baojiang
 Zhou, Cathy
 Zhou, Changyin
 Zhou, Feng
 Zhou, Jun
 Zhou, S.
 Zhu, Feng
 Zou, Danping
 Zucker, Steve

Additional Reviewers

Bai, Xiang	Liu, Damon Shing-Min
Collins, Toby	Liu, Dong
Compte, Benot	Luo, Ye
Cong, Yang	Magerand, Ludovic
Das, Samarjit	Molineros, Jose
Duan, Lixing	Rao, Shankar
Fihl, Preben	Samir, Chafik
Garro, Valeria	Sanchez-Riera, Jordy
Geng, Bo	Suryanarayana, Venkata
Gherardi, Riccardo	Tang, Sheng
Giusti, Alessandro	Thota, Rahul
Guo, Jing-Ming	Toldo, Roberto
Gupta, Vipin	Tran, Du
Han, Long	Wang, Jingdong
Korchev, Dmitriy	Wu, Jun
Kulkarni, Kaustubh	Yang, Jianchao
Lewandowski, Michal	Yang, Linjun
Li, Xin	Yang, Kuiyuan
Li, Zhu	Yuan, Fei
Lin, Guo-Shiang	Zhang, Guofeng
Lin, Wei-Yang	Zhuang, Jinfeng

ACCV2010 Best Paper Award Committee

Alfred M. Bruckstein	Technion, Israel Institute of Technology, Israel
Larry S. Davis	University of Maryland, USA
Richard Hartley	Australian National University, Australia
Long Quan	The Hong Kong University of Science and Technology, Hong Kong

Sponsors of ACCV2010

Main Sponsor	The Asian Federation of Computer Vision Societies (AFCV)
Gold Sponsor	NextWindow – Touch-Screen Technology
Silver Sponsors	Areograph – Interactive Computer Graphics Microsoft Research Asia Australia’s Information and Communications Technology (NICTA) Adept Electronic Solutions
Bronze Sponsor	4D View Solutions
Best Student Paper Sponsor	<i>The International Journal of Computer Vision</i> (IJCV)

Best Paper Prize ACCV 2010

Context-Based Support Vector Machines for Interconnected Image Annotation
Hichem Sahbi, Xi Li.

Best Student Paper ACCV 2010

Fast Spectral Reflectance Recovery Using DLP Projector
Shuai Han, Imari Sato, Takahiro Okabe, Yoichi Sato

Best Application Paper ACCV 2010

Network Connectivity via Inference Over Curvature-Regularizing Line Graphs
Maxwell Collins, Vikas Singh, Andrew Alexander

Honorable Mention ACCV 2010

Image-Based 3D Modeling via Cheeger Sets
Eno Toeppe, Martin Oswald, Daniel Cremers, Carsten Rother

Outstanding Reviewers ACCV 2010

Philippos Mordohai
Peter Roth
Matt Toews
Andres Bruhn
Sudipta Sinha
Benjamin Berkels
Mathieu Salzmann

Table of Contents – Part IV

Posters on Day 3 of ACCV 2010

Fast Computation of a Visual Hull	1
<i>Sujung Kim, Hee-Dong Kim, Wook-Joong Kim, and Seong-Dae Kim</i>	
Active Learning with the Furthest Nearest Neighbor Criterion for Facial Age Estimation	11
<i>Jian-Gang Wang, Eric Sung, and Wei-Yun Yau</i>	
Real-Time Human Detection Using Relational Depth Similarity Features	25
<i>Sho Ikemura and Hironobu Fujiyoshi</i>	
Human Tracking by Multiple Kernel Boosting with Locality Affinity Constraints	39
<i>Fan Yang, Huchuan Lu, and Yen-Wei Chen</i>	
A Temporal Latent Topic Model for Facial Expression Recognition	51
<i>Lifeng Shang and Kwok-Ping Chan</i>	
From Local Features to Global Shape Constraints: Heterogeneous Matching Scheme for Recognizing Objects under Serious Background Clutter	64
<i>Martin Klinkigt and Koichi Kise</i>	
3D Structure Refinement of Nonrigid Surfaces through Efficient Image Alignment	76
<i>Yinqiang Zheng, Shigeki Sugimoto, and Masatoshi Okutomi</i>	
Local Empirical Templates and Density Ratios for People Counting	90
<i>Dao Huu Hung, Sheng-Luen Chung, and Gee-Sern Hsu</i>	
Curved Reflection Symmetry Detection with Self-validation	102
<i>Jingchen Liu and Yanxi Liu</i>	
An HMM-SVM-Based Automatic Image Annotation Approach	115
<i>Yinjie Lei, Wilson Wong, Wei Liu, and Mohammed Bennamoun</i>	
Video Deblurring and Super-Resolution Technique for Multiple Moving Objects	127
<i>Takuma Yamaguchi, Hisato Fukuda, Ryo Furukawa, Hiroshi Kawasaki, and Peter Sturm</i>	

Sparse Source Separation of Non-instantaneous Spatially Varying Single Path Mixtures	141
<i>Albert Achtenberg and Yehoshua Y. Zeevi</i>	
Improving Gaussian Process Classification with Outlier Detection, with Applications in Image Classification	153
<i>Yan Gao and Yiqun Li</i>	
Robust Tracking Based on Pixel-Wise Spatial Pyramid and Biased Fusion	165
<i>Huchuan Lu, Shipeng Lu, and Yen-Wei Chen</i>	
Compressive Evaluation in Human Motion Tracking	177
<i>Yifan Lu, Lei Wang, Richard Hartley, Hongdong Li, and Dan Xu</i>	
Reconstructing Mass-Conserved Water Surfaces Using Shape from Shading and Optical Flow	189
<i>David Pickup, Chuan Li, Darren Cosker, Peter Hall, and Phil Willis</i>	
Earth Mover’s Morphing: Topology-Free Shape Morphing Using Cluster-Based EMD Flows	202
<i>Yasushi Makihara and Yasushi Yagi</i>	
Object Detection Using Local Difference Patterns	216
<i>Satoshi Yoshinaga, Atsushi Shimada, Hajime Nagahara, and Rin-ichiro Taniguchi</i>	
Randomised Manifold Forests for Principal Angle-Based Face Recognition	228
<i>Ujwal D. Bonde, Tae-Kyun Kim, and Kalpatti R. Ramakrishnan</i>	
Estimating Meteorological Visibility Using Cameras: A Probabilistic Model-Driven Approach	243
<i>Nicolas Hautière, Raouf Babari, Éric Dumont, Roland Brémond, and Nicolas Paparoditis</i>	
Optimizing Visual Vocabularies Using Soft Assignment Entropies	255
<i>Yubin Kuang, Kalle Åström, Lars Kopp, Magnus Oskarsson, and Martin Byröd</i>	
Totally-Corrective Multi-class Boosting	269
<i>Zhihui Hao, Chunhua Shen, Nick Barnes, and Bo Wang</i>	
Pyramid Center-Symmetric Local Binary/Trinary Patterns for Effective Pedestrian Detection	281
<i>Yongbin Zheng, Chunhua Shen, Richard Hartley, and Xinsheng Huang</i>	

Reducing Ambiguity in Object Recognition Using Relational Information	293
<i>Kuk-Jin Yoon and Min-Gil Shin</i>	
Posing to the Camera: Automatic Viewpoint Selection for Human Actions	307
<i>Dmitry Rudoy and Lihí Zelnik-Manor</i>	
Orthogonality Based Stopping Condition for Iterative Image Deconvolution Methods	321
<i>Dániel Szolgay and Tamás Szirányi</i>	
Probabilistic 3D Object Recognition Based on Multiple Interpretations Generation	333
<i>Zhaojin Lu, Sukhan Lee, and Hyunwoo Kim</i>	
Planar Affine Rectification from Change of Scale	347
<i>Ondřej Chum and Jiří Matas</i>	
Sensor Measurements and Image Registration Fusion to Retrieve Variations of Satellite Attitude	361
<i>Régis Perrier, Elise Arnaud, Peter Sturm, and Mathias Ortner</i>	
Image Segmentation Fusion Using General Ensemble Clustering Methods	373
<i>Lucas Franek, Daniel Duarte Abdala, Sandro Vega-Pons, and Xiaoyi Jiang</i>	
Real Time Myocardial Strain Analysis of Tagged MR Cines Using Element Space Non-rigid Registration	385
<i>Bo Li, Brett R. Cowan, and Alistair A. Young</i>	
Extending AMCW Lidar Depth-of-Field Using a Coded Aperture	397
<i>John P. Godbaz, Michael J. Cree, and Adrian A. Dorrington</i>	
Surface Extraction from Iso-disparity Contours	410
<i>Chris McCarthy and Nick Barnes</i>	
Image De-fencing Revisited	422
<i>Minwoo Park, Kyle Brocklehurst, Robert T. Collins, and Yanxi Liu</i>	
Feature-Assisted Dense Spatio-temporal Reconstruction from Binocular Sequences	435
<i>Yihao Zhou and Yan Qiu Chen</i>	
Improved Spatial Pyramid Matching for Image Classification	449
<i>Mohammad Shahiduzzaman, Dengsheng Zhang, and Guojun Lu</i>	
Dense Multi-frame Optic Flow for Non-rigid Objects Using Subspace Constraints	460
<i>Ravi Garg, Luis Pizarro, Daniel Rueckert, and Lourdes Agapito</i>	

Fast Recovery of Weakly Textured Surfaces from Monocular Image Sequences	474
<i>Oliver Ruepp and Darius Burschka</i>	
Ghost-Free High Dynamic Range Imaging	486
<i>Yong Seok Heo, Kyoung Mu Lee, Sang Uk Lee, Youngsu Moon, and Joonhyuk Cha</i>	
Pedestrian Recognition with a Learned Metric	501
<i>Mert Dikmen, Emre Akbas, Thomas S. Huang, and Narendra Ahuja</i>	
A Color to Grayscale Conversion Considering Local and Global Contrast	513
<i>Jung Gap Kuk, Jae Hyun Ahn, and Nam Ik Cho</i>	
Affordance Mining: Forming Perception through Action	525
<i>Liam Ellis, Michael Felsberg, and Richard Bowden</i>	
Spatiotemporal Contour Grouping Using Abstract Part Models	539
<i>Pablo Sala, Diego Macrini, and Sven Dickinson</i>	
Efficient Multi-structure Robust Fitting with Incremental Top- k Lists Comparison	553
<i>Hoi Sim Wong, Tat-Jun Chin, Jin Yu, and David Suter</i>	
Flexible Online Calibration for a Mobile Projector-Camera System	565
<i>Daisuke Abe, Takayuki Okatani, and Koichiro Deguchi</i>	
3D Object Recognition Based on Canonical Angles between Shape Subspaces	580
<i>Yosuke Igarashi and Kazuhiro Fukui</i>	
An Unsupervised Framework for Action Recognition Using <i>Actemes</i>	592
<i>Kaustubh Kulkarni, Edmond Boyer, Radu Horaud, and Amit Kale</i>	
Segmentation of Brain Tumors in Multi-parametric MR Images via Robust Statistic Information Propagation	606
<i>Hongming Li, Ming Song, and Yong Fan</i>	
Face Recognition with Decision Tree-Based Local Binary Patterns	618
<i>Daniel Maturana, Domingo Mery, and Álvaro Soto</i>	
Occlusion Handling with ℓ_1 -Regularized Sparse Reconstruction	630
<i>Wei Li, Bing Li, Xiaoqin Zhang, Weiming Hu, Hanzhi Wang, and Guan Luo</i>	
An Approximation Algorithm for Computing Minimum-Length Polygons in 3D Images	641
<i>Fajie Li and Xiuxia Pan</i>	

Classifier Acceleration by Imitation	653
<i>Takahiro Ota, Toshikazu Wada, and Takayuki Nakamura</i>	
Recognizing Continuous Grammatical Marker Facial Gestures in Sign Language Video	665
<i>Tan Dat Nguyen and Surendra Ranganath</i>	
Invariant Feature Set Generation with the Linear Manifold Self-organizing Map	677
<i>Huicheng Zheng</i>	
A Multi-level Supporting Scheme for Face Recognition under Partial Occlusions and Disguise	690
<i>Jacky S-C. Yuk, Kwan-Yee K. Wong, and Ronald H-Y. Chung</i>	
Foreground and Shadow Segmentation Based on a Homography-Correspondence Pair	702
<i>Haruyuki Iwama, Yasushi Makihara, and Yasushi Yagi</i>	
Author Index	717

Fast Computation of a Visual Hull

Sujung Kim, Hee-Dong Kim, Wook-Joong Kim, and Seong-Dae Kim

Department of Electrical Engineering,
Korea Advanced Institute of Science and Technology, Korea

Abstract. Two techniques for the fast computation of a visual hull without simplification are proposed. First, we tackle the most time consuming step for finding the intersections between projected rays and silhouette boundaries. We use the chain coding representation of silhouette boundaries for fast searching and computing with sub-pixel accuracy. Second, we analyze 3D-2D projection and back-projection relations and formulate them as 1D homographies. This formulation reduces computational cost and ambiguity that can be caused by measurement errors in the back-projection of 2D intersections to 3D. Furthermore, we show that the formulation is not limited to the projective space but also useful in the affine space. We generalize our techniques to an arbitrary 3D ray, so that the proposed method is directly applicable to both volume-based and surface-based visual hull methods. In our simulations, we compare the proposed algorithm with the state-of-the-art methods and show its advantages in terms of computational cost.

1 Introduction

A visual hull is a three-dimensional (3D) boundary constructed by the silhouettes of an object from multiple views. Visual hull computing is relatively simple and robust, and so it is commonly used in various applications such as image-based modeling, real-time geometry capture systems, and gesture recognition. In [7], Laurentini et al. first defined a visual hull and presented its fundamental properties. Since then, a number of studies have been devoted for obtaining geometric details about a visual hull in efficient ways.

Generally, visual hull computing is classified into two types: volume-based approach and surface-based approach. Volume-based approaches, which are traditional, consider a visual hull as the union of voxels whose projections are inside 2D silhouettes. These approaches are straightforward to implement but voxel resolution has to be increased for building a precise visual hull, which consequently leads to a heavy computational burden. In practice, the computational complexity of voxel carving algorithm is $O(n^3)$ where n is the resolution of single dimension. To reduce computational cost, a number of schemes using octree data structure have been developed to efficiently compute a visual hull [3, 13]. In [9, 11, 14], the complexity issue was tackled by modifying voxel-wise computation into the intersection problem of line-to-silhouette boundaries. These are, however, still heavy to compute a precise object for real time applications.

Recently, for speeding up the computational time, computations in a volume-based approach are effectively parallelized and implemented on the GPU [6,14].

In comparison, surface-based approaches compute a visual hull with the intersections of visual cones which are the back-projections of silhouettes in the 3D space. These approaches can provide more detailed information about original surfaces. However, when the shape of an object is complicated, it is likely to yield numerical instability due to a degenerate case or the singularity problem. In [1,10], to resolve the instability, 3D intersections and surfaces were computed on the 2D domain based on epipolar geometry. Additionally, they approximated silhouette boundaries as polygons to reduce a computational cost, but such simplification makes a final visual hull lose details of shapes. Lazebnik et al. [8] proposed an image-based method to represent a visual hull without simplification, but it is impractical for fast and robust computations. Recently, Franco et al. [2] presented a simpler alternative to compute a visual hull with pixel-exact boundaries. Even though it improved previous results in terms of geometric precision as well as computational efficiency, it requires considerable computational time to estimate a visual hull with sub-pixel contours.

To determine the performance of visual hull computing algorithms, geometric accuracy and computational efficiency are two crucial aspects. So far, most methods have mainly focused on obtaining detailed geometric information about a visual hull. When it comes to a computation cost, they attempted to simplify the representation of a visual hull through lowering voxel resolution or approximating silhouette boundaries. However, as we can easily expect, such simplification makes a final visual hull include unnecessary parts or lose details of shapes. Recently, to meet both goals which are geometric accuracy and computational efficiency, higher voxel resolution and sub-pixel silhouette boundaries are used. Hence, in order to improve computational efficiency without simplification, it is necessary to reduce the most consuming part in computation of a visual hull.

In this paper, we propose a fast visual hull computing algorithm without simplification on a visual hull. In the overall process of visual hull computing, the step for finding the intersections of 3D rays and 2D silhouettes is the most time consuming part. The proposed algorithm tackles this step by following two techniques: first, we use the chain-coding representation for fast searching of the intersections with sub-pixel accuracy; second, we analyze the projection (3D-to-2D) and back-projection (2D-to-3D), and formulate them as line (1D) homographies. In this formulation, we can avoid ambiguity in back-projecting 2D intersections to 3D and also can achieve compact computations. This paper is organized as follows: Section 2 introduces the representation of silhouettes using the chain codes and presents the fast search method of intersections. In Section 3, we describe the proposed line homography techniques. In Section 4, experimental results are presented, and finally, we conclude in Section 5.

2 The Proposed Method

In computing a visual hull, for given a single 3D ray, following steps are usually required: 1) projection of a ray onto a silhouette image; 2) computation

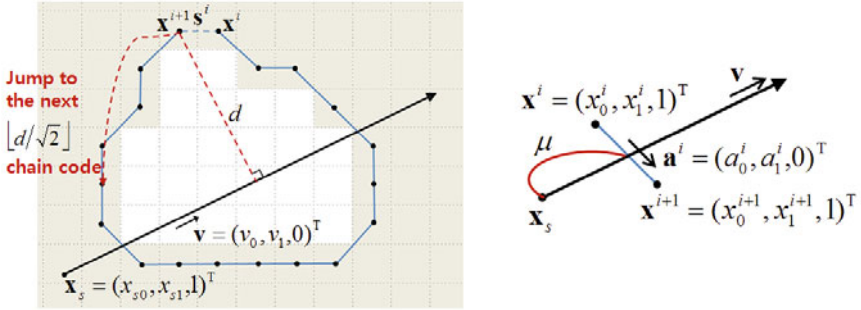


Fig. 1. Computing intersections based on chain coding representation of silhouette boundaries

of a line segment connecting the intersections between a projected ray and silhouette boundaries; and 3) transformation of the line segment to a 3D ray by back-projection. Among these steps, we reduce the complexity of step 2), which is the most time consuming, by chain-coding representation. Chain-coding representation is one of the fundamental processes for binary image processing and is computationally cheap. In step 2), we only use chain codes instead of preprocessing such as rectifications [1] or edge bins according to contour points [10]. The proposed method has a lower incremental rate of computational complexity compared to previous methods when image resolution is increased. In addition, we generalize this technique to an arbitrary 3D ray, so that it is also useful in a volume-based approach which is improper to use epipolar constraints. Furthermore, in step 1) and 3) we formulate the 3D and 2D relations as 1D line homographies to alleviate inaccurate computation that might be caused by camera parameter errors or measurement noises. This interpretation leads to the reduction of a computational cost.

2.1 Computation of Intersections

A chain code is a data structure which defines a regional boundary (contour) by means of eight (or four) directional vectors, and chain code representation is the process for generating a sequence of chain codes based on the neighborhood relationship among adjacent boundary pixels. In our algorithm, we use the chain coding representation of [4], which contains a start code so that a closed contour can be represented by multiple segments, and traces internal and external boundaries in clockwise and counter-clockwise directions, respectively.

Chain coding representation of silhouette boundaries can bring following two benefits: first, it enables the computation of the intersections between silhouettes and projected rays with sub-pixel precision because the intersection can be considered as a line-crossing problem between a projected 3D ray and a chain-code vector. Second, since the location of a boundary pixel is identified simply from a corresponding chain code, we can easily exclude irrelevant pixels

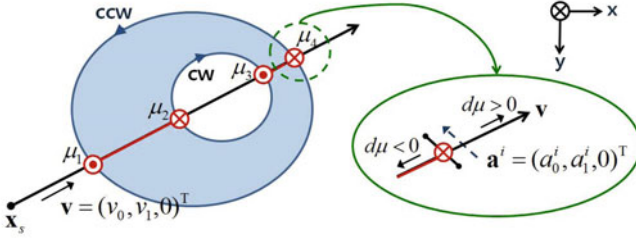


Fig. 2. Validity test: valid line segments of a target object are found using the cross product of directional vectors of a chain code and a projected ray

for fast computing. For instance, if a boundary pixel is far enough from a projected ray, we can ignore neighboring pixels of it because these are unlikely to be near an intersection point.

Now we describe the chain code based method for finding ray-to-silhouette intersections. Assume that \mathbf{l} is the projection of a 3D ray onto an image plane containing a silhouette \mathbf{S} . Line \mathbf{l} is determined by its start point $\mathbf{x}_s = (x_{s0}, x_{s1}, 1)^T$ and directional vector $\mathbf{v} = (v_0, v_1, 0)^T$, and silhouette \mathbf{S} is represented as a sequence of N chain codes as $\mathbf{S} = \bigcup_{i=1}^N \mathbf{s}^i$, where \mathbf{s}^i is the i^{th} chain code. Given the chain codes and the projected ray, first, we conduct a crossing test which identifies the intersection between \mathbf{l} and \mathbf{S} . For the crossing test, we compute the signed distances between \mathbf{l} and two neighboring pixels constituting \mathbf{s}^i as:

$$C = \{(v_1 x_0^i - v_0 x_1^i) - (v_1 x_{s0} - v_0 x_{s1})\} \{(v_1 x_0^{i+1} - v_0 x_1^{i+1}) - (v_1 x_{s0} - v_0 x_{s1})\} \quad (1)$$

where, $\mathbf{x}^i = (x_0^i, x_1^i, 1)^T$ and $\mathbf{x}^{i+1} = (x_0^{i+1}, x_1^{i+1}, 1)^T$ are the corresponding pixels to \mathbf{s}^i . If \mathbf{s}^i intersects with \mathbf{l} , C becomes negative (otherwise, C is positive).

Meanwhile, in the crossing test, it is unnecessary to compute the signed distance for every single chain code. As illustrated in Fig. 2(left), assume that the shortest Euclidean distance from x^{i+1} of a chain code \mathbf{s}^i to \mathbf{l} is $d (> 0)$. Since the possible maximum distance between neighboring pixels is $\sqrt{2}$ (i.e., Euclidean distance between two diagonal pixels), following $\lfloor d/\sqrt{2} \rfloor$ chain codes never intersect with \mathbf{l} , which subsequently accelerates scanning speed by deleting the irrelevant chain codes. In our experiments, more than 96% of chain codes were excluded in the crossing test.

Once a chain code \mathbf{s}^i is determined to meet with \mathbf{l} using (1), next, we compute the location of an intersection point. In this computing, instead of finding the point in an image grid, we compute distance μ from starting point $\mathbf{x}_s = (x_{s0}, x_{s1}, 1)^T$ on line \mathbf{l} as:

$$\mu = \frac{a_1^i(x_0^i - x_{s0}) - a_0^i(x_1^i - x_{s1})}{(a_1^i v_0 - a_0^i v_1)} \quad (2)$$

where, as in Fig. 2(right), $\mathbf{a}^i = (a_0^i, a_1^i, 0)^T$ is the directional vector of \mathbf{s}^i . After the intersections are found, finally, we conduct the validity test to find valid

line segments which construct a visual hull. Fig. 2 illustrates this test by an example. Given four intersection points on a projected ray (circles with dots or crosses inside), a projected ray can be divided into five line segments. Among them, only two segments correspond to an object region whereas others have to be eliminated. Due to chain coding, the validity of line segments can be simply identified by computing the cross product of the directional vectors of an intersecting chain code and a projected line as:

$$\mathbf{a}^i \times \mathbf{v} = \begin{vmatrix} i & j & k \\ a_0^i & a_1^i & 0 \\ v_0 & v_1 & 0 \end{vmatrix} = \begin{bmatrix} 0 \\ 0 \\ a_0^i v_1 - a_1^i v_0 \end{bmatrix} \quad (3)$$

Since the external boundary and the internal boundary of a silhouette are coded in counter-clockwise and clockwise directions, respectively, silhouette regions for a target object must be located on the left side of the boundaries. Therefore, as illustrated inside of green circle in Fig. 2, if $a_0^i v_1 - a_1^i v_0$ is negative (circle with a cross), a target object lies on the decreasing direction of a projected ray (represented as the red segment, i.e., $d\mu < 0$), otherwise, a target object on the increasing direction (i.e., $d\mu > 0$).

2.2 1D Line Homography for 3D Ray Back-Projection

Generally, the projection (3D ray to 2D line) and its back-projection (2D line to 3D ray) are computed using a camera projection matrix (or epipolar geometry) and its pseudo-inverse respectively. In practice, such computation might cause numerical instability due to camera parameter errors or measurement noises. Especially, since the estimation of back-projected 3D rays, \mathbf{B}_i in Fig. 3, is susceptible to even small errors in measurement noises, the crossing points of \mathbf{B}_i and \mathbf{L} often become ambiguous. To resolve this computational instability, we analyze the 2D and 3D relations, and present that those relations can be formulated as 1D line homographies. Such interpretation can bring not only compact computation but also computational robustness in finding a 3D ray intersection.

Given a 3D ray defined by a start 3D point $\mathbf{X}_0 = (X_{o0}, X_{o1}, X_{o2}, 1)^T$ and a directional vector $\mathbf{D} = (D_0, D_1, D_2, 0)^T$, 3D point \mathbf{X} on the 3D ray can be defined as $\mathbf{X}_0 + \lambda \mathbf{D}$, where λ represents the distance from \mathbf{X}_0 along a 3D ray. For a given camera having its projection matrix $\mathbf{P} = [\mathbf{M}|\mathbf{t}]$, the projected point \mathbf{x} of \mathbf{X} (i.e., $\mathbf{x} = \mathbf{P}\mathbf{X}$) is represented as $\mathbf{x} = \mathbf{x}_0 + \lambda \mathbf{d}$, where $\mathbf{x}_0 = \mathbf{P}\mathbf{X}_0$ and $\mathbf{d} = (d_0, d_1, d_2)^T$ is from $\mathbf{d} = \mathbf{M}\mathbf{D}$, where $\mathbf{D} = (D_0, D_1, D_2)^T$.

According to the oriented projective geometry theorem [12], if x_{o2} of $\mathbf{x}_0 = (x_{o0}, x_{o1}, x_{o2})^T$ is negative when $\mathbf{P} = [\mathbf{M}|\mathbf{t}]$ and \mathbf{X}_0 are oriented, \mathbf{x}_0 cannot be found on an image plane. Hence, to avoid this invisibility issue, we define an arbitrary visible point \mathbf{x}_v as $\mathbf{x}_v = \mathbf{x}_o + \lambda_v \mathbf{d}$. Then we force $x_{o2} + \lambda_v d_2$ to be always positive (Here, we set $x_{o2} + \lambda_v d_2$ by 1 for the simple representation of 1D homography), so that we can measure the distance corresponding to λ on a silhouette image. Representing \mathbf{x} with \mathbf{x}_v , we have

$$\mathbf{x} = \mathbf{x}_v + (\lambda - \lambda_v) \mathbf{d} \quad (4)$$

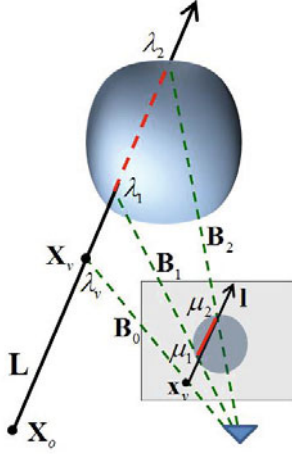


Fig. 3. The relationship between a 3D ray and a projected ray on a silhouette image

However, since \mathbf{d} might not be the direction vector of a projected ray \mathbf{l} containing \mathbf{x} , we modify (4) using directional vector $\mathbf{v} = (v_0, v_1, 0)^T$ as

$$\mathbf{x} = \mathbf{x}_v + \mu \mathbf{v} \tag{5}$$

where μ is the distance from a visible point \mathbf{x}_v along \mathbf{l} . Note that $\mathbf{x}_v = (x_{v0}, x_{v1}, 1)^T$ and μ are measurable on the image plane. For obtaining \mathbf{v} from (4), we differentiate it by λ and hence \mathbf{v} is represented as

$$\begin{bmatrix} v_0 \\ v_1 \end{bmatrix} \equiv \begin{bmatrix} \partial x / \partial \lambda \\ \partial y / \partial \lambda \end{bmatrix} \Big|_{\lambda=\lambda_v} = \begin{bmatrix} d_0 x_{o2} - d_2 x_{o0} \\ d_1 x_{o2} - d_2 x_{o1} \end{bmatrix} \tag{6}$$

Now, we have all elements in (4) and (5), and so we can obtain the relations between λ and μ as (7) using directional vector (6) and the distance between \mathbf{x}_v and \mathbf{x} in (4).

$$\mu = \frac{\lambda - \lambda_v}{x_{o2} + \lambda d_2} \tag{7}$$

Finally, we formulate (7) in the form of line (1D) homographies as:

$$\begin{bmatrix} k\mu \\ k \end{bmatrix} = \begin{bmatrix} 1 & -\lambda_v \\ d_2 & x_{o2} \end{bmatrix} \begin{bmatrix} \lambda \\ 1 \end{bmatrix} \tag{8}$$

$$\begin{bmatrix} k\lambda \\ k \end{bmatrix} = \begin{bmatrix} x_{o2} & \lambda_v \\ -d_2 & 1 \end{bmatrix} \begin{bmatrix} \mu \\ 1 \end{bmatrix} \tag{9}$$

(8) and (9) are projection(from 3D to 2D) and back-projection(from 2D to 3D) in the form of 1D homographies respectively. Note that for back-projecting a 2D intersection point to 3D, instead of computing pseudo-inverse of camera



Fig. 4. Input datasets. From left to right: temple, dino, alien, dinosaur, and predator. (top) a sample input image from each dataset, (bottom) reconstructions of the proposed method in a surface-based approach.

projection matrix, we only compute last elements of \mathbf{x}_0 and \mathbf{d} (x_{o2} and d_2), and a visible distance λ_v can be computed by $x_{o2} + \lambda_v d_2 = 1$. Furthermore, in the affine space, we can obtain simpler version of the 3D-to-2D and the 2D-to-3D projections as (10) and (11). The third row of a affine camera matrix \mathbf{P} is represented by $\mathbf{p}^{3T} = (0, 0, 0, 1)^T$, so that the last elements of \mathbf{x}_0 and \mathbf{d} are one and zero respectively, and finally it results in the identity matrix for transformations between 2D and 3D in the affine space.

$$\begin{bmatrix} k\mu \\ k \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \lambda \\ 1 \end{bmatrix} \quad (10)$$

$$\begin{bmatrix} k\lambda \\ k \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mu \\ 1 \end{bmatrix} \quad (11)$$

Therefore, with (8) and (9) for the projective space, and with (10) and (11) for the affine space, we can simplify the process of mapping between 3D and 2D while avoiding the ambiguity of intersection points.

3 Experimental Results

We tested our proposed method on two data sets containing multiview calibrated images: the first data set [5] is composed of two sequences, *temple* and *dino*, 16 views each, and has the resolution of 640x480; and the second data set [8] contains 24 views of *alien*(1600x1600), *dinosaur*(2000x1500) and *predator*(1800x1700). Fig 4 shows one of input images in each data set and the corresponding reconstructed 3D points using the proposed method in a surface-based approach.

First, we compared the computational time with the method for finding intersections in [2], which is recent and widely used. All the experiments are conducted

Table 1. Comparisons of computational time on a surface-based approach: the second column represents the average number of contour points per image, and last two columns show the running times of both methods

	Average contour pts.	Vertices	Method in [2]	Proposed method
Temple	1995	38254	2.866	2.118
Dino	1484	39212	2.838	1.716
Alien	6450	114202	62.578	17.572
Dinosaur	5337	150948	47.165	14.164
Predator	7277	211414	84.471	21.296

Table 2. Comparisons of computational time with a conventional voxel carving method

	running time(sec)			
	Voxel carving	Proposed method	Voxel carving	Proposed method
Voxel size	256x256x256		512x512x512	
Temple	90.064	1.188	712.610	4.375
Dino	94.767	1.282	714.687	4.750
Alien	88.188	1.281	701.750	4.625
Dinosaur	88.859	1.218	705.172	4.484
Predator	88.500	1.312	705.219	4.813

with a 2.67 GHz Intel Core i5 processor and 3 GB of RAM. As Table 1 shows, our method outperforms the method in [2] on both data sets. And especially for the second data set which has higher resolution, the computational gain was about three-times higher than the previous method. Since the complexity of our method is not proportional to the number of contour pixels (i.e., edge bins in [2]), it is likely that the higher resolution images we have, the better computational efficiency we obtain.

Second, we applied the proposed method to a volume-based approach. Recently, Lee et al [9] proposed an efficient way to reduce a computational cost by modifying voxel-wise computation into the intersection problem of line-to-silhouette boundaries. We combined our method with Lees and compared computational times with a conventional voxel carving method. Table 2 shows its results. Though no optimization was used in the conventional voxel carving, the computational time was drastically reduced down to 1% compared to the conventional method. In practice, an octree based volume representation is widely used for a real time rendering system on the GPU. When the computational time of the conventional voxel carving algorithm is N^3 where, N is the resolution of single dimension, that of the octree based volume representation is computed by $\sum_{i=0}^{N-1} 8 \cdot (8q)^i$ where, q is an accept ratio. If an accept ratio is 0.7 on each octree, the computational time of the octree based method is decreased to 7% of that of the conventional method, but it is still higher than the result of the proposed method. Recently, in [14] the computational time is significantly reduced using pre-defined bins which are the modification of [10]. As shown in the experiments

of a surface-based approach, even though using pre-defined edge bins can accelerate to find intersections, our method outperforms it, especially when image resolution is higher. Additionally, since the computation of our method can be perfectly parallelized, it is well suited to be implemented on the GPU. Hence we can presume that the proposed method can be also used effectively for a volume-based approach.

4 Conclusion

We presented the fast computation algorithm of a visual hull using chain coding representation of silhouette boundaries. Additionally, we formulated the relations between 3D and 2D (projection and back-projection) as 1D line homographies, so that we were able to improve computational resilience to noises. The proposed method is directly applicable to both methods for computing a visual hull since we generalized our method to an arbitrary 3D ray. In experiments, we showed that the proposed method is computationally efficient compared to the recent method in [2]. In the future, we will implement our method on the GPU. We believe that our techniques will be useful tools for speeding up a real time 3D reconstruction system.

References

1. Boyer, E., Franco, J.: A Hybrid Approach for Computing Visual Hulls of Complex Objects. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, vol. 1, pp. 695–701 (2003)
2. Boyer, E., Franco, J.: Efficient Polyhedral Modeling from Silhouettes. *IEEE Trans. Pattern Analysis and Machine Intelligence* 31, 414–427 (2009)
3. Cheung, G., Kanade, T., Bouguet, J., Holler, M.: A real time system for robust 3D voxel reconstruction of human motions. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 714–720 (2000)
4. Freeman, H.: Computer Processing of Line-Drawing Images. *ACM Computing Surveys* 6, 57–97 (1974)
5. Goesele, M., Curless, B., Seitz, S.: Multi-View Stereo Revisited. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, vol. 1, pp. 564–577 (2006)
6. Ladikos, A., Benhimane, S., Navab, N.: Efficient visual hull computation for real-time 3D reconstruction using CUDA. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops (2008)
7. Laurentini, A.: The Visual Hull Concept for Silhouette-Based Image Understanding. *IEEE Trans. Pattern Analysis and Machine Intelligence* 16, 150–162 (1994)
8. Lazebnik, S., Furukawa, Y., Ponce, J.: Projective Visual Hulls. *Int. J. Computer Vision* 74, 137–165 (2007)
9. Lee, J., Rhee, K., Park, S., Kim, S.: Efficient three-dimensional object representation and reconstruction using depth and texture maps. *Optical Engineering* 47, 17204 (2008)
10. Matusik, W., Buehler, C., Raskar, R., Gortler, S., McMillan, L.: Image Based Visual Hulls. In: Proc. ACM SIGGRAPH 2000, ACM Computer Graphics, pp. 369–374 (2000)

11. Niem, W.: Automatic Modeling of 3D Natural Objects from Multiple Views. In: Proc. European Workshop Combined Real and Synthetic Image Processing for Broadcast and Video Production (1994)
12. Stolfi, J.: Oriented Projective Geometry: A Framework for Geometric Computations. Academic Press, London (1991)
13. Szeliski, R.: Rapid octree construction from image sequences. *CVGIP: Image understanding* 58, 23–32 (1993)
14. Waizenegger, W., Feldmann, I., Eisert, P., Kauff, P.: Parallel high resolution real-time visual hull on GPU. In: Proc. IEEE International Conference of Image Processing, pp. 4301–4304 (2009)

Active Learning with the Furthest Nearest Neighbor Criterion for Facial Age Estimation

Jian-Gang Wang¹, Eric Sung², and Wei-Yun Yau¹

¹ Institute for Infocomm Research, 1 Fusionopolis Way
#21-01 Connexis, Singapore 138632

² Nanyang Technological University, Singapore 639798

Abstract. Providing training data for facial age estimation is very expensive in terms of age progress, privacy, human time and effort. In this paper, we present a novel active learning approach based on an on-line Two-Dimension Linear Discriminant Analysis for learning to quickly reach high performance but with minimal labeling effort. The proposed approach uses the classifier learnt from the small pool of labeled faces to select the most informative samples from the unlabeled set to increasingly improve the classifier. Specifically, we propose a novel data selection of the Furthest Nearest Neighbour (FNN) that generalizes the margin-based uncertainty to the multi-class case and which is easy to compute so that the proposed active learning can handle a large number of classes and large data sizes efficiently. Empirical experiments on FG-NET, Morph databases and a large unlabeled data set show that the proposed approach can achieve similar results using fewer samples than random selection.

1 Introduction

Facial age estimation is an approach to classify images into one of several pre-defined age-groups. There are many applications of facial age estimation, e.g. digital signage in which the statistics of peoples' age ranges can provide reference to manage the advertising. Another interesting application is for interactive games. An intelligent toy can perform different games based on the player's age range, e.g. it can automatically provide easy games such as popular crossword puzzles or board games when an elderly person is detected and card games, video games and computer games when a teenager is detected. One of the difficulties in age estimation using face images is that the training database is highly incomplete. In order to collect photos of a person, the subject could be required to scan his/her photos captured during the past at his/her different age. On the other hand, there are a lot of unlabeled face images. Although the age range can be roughly estimated by humans from a face image, labeling such large data set is very time consuming. Furthermore, there is possibility of incorrect labeling due to the subjective nature of the observer, the quality of the face images, the viewpoint, scenery, familiarity and the fact that there are people whose look defies their age. Hence, in this paper, we hope to provide a framework to reduce the

amount of labeling effort needed by selecting the most informative face images to be labeled.

Recent research in the area of active learning [2, 4, 7, 13, 18, 20, 26] has been reasonably successful in handling the problem of active selection of the examples to be labeled. The purpose of active learning is to minimize the selection of samples from the unlabeled pool to be labeled by the oracle in order to fully learn the complete data available. The idea for selecting a sample is that the worst samples (with the biggest error) should be added to the training samples and a new classifier will be learned using the new training database. However, as the data is unlabeled, one cannot tell which is the violating data. In the conventional methods using SVM classifiers, the selection of unlabeled data for further learning are the ones nearest to the current optimal hyperplane [28]. In addition, when both the data set and the dimension of the feature vector become very large, training with the complete data set is infeasible. Hence, it is clear that an incremental (sequential) learning scheme is needed for the sample selection and updates the discriminant eigenspace with light computation instead of full re-training. Therefore, we will combine the two tasks of active learning and sequential learning. In this paper, a novel Incremental Two-dimensional Linear Discriminant Analysis (I2DLDA) with active learning is proposed to classify face image into one of several age categories. The key issue here is that for the LDA approach, unlike other methods [8, 28], the notion of selecting the next most informative image for labeling has not been examined. For our proposed 2DLDA with nearest neighbour classifier, the unlabeled data with the most uncertainty will have to be measured and chosen differently. In principle, they are the data which are the furthest among the nearest neighbours. For the cases where the selection turns out to be correctly classified by the current 2DLDA classifier or even where it is wrongly classified, it will give the highest probability of generalization. We propose here to apply the measure of choosing the Furthest Nearest Neighbour (FNN). The rationale and the experimental verification that this is indeed a viable measure will be dealt with in detail later in the paper.

So far there is relatively few literature on automatic age estimation [5, 11, 12, 23, 32] compared to other facial processing such as face recognition and facial gender recognition. A recent survey can be found in [24]. The related work closest to our approach is that by Gen et al. [6] who noticed the incomplete data problem and proposed an aging pattern subspace, named AGES (AGing pattErn Subspace), for estimating age from appearance. In order to handle incomplete data such as missing ages in the training sequence, the AGES method models a sequence of individual aging face images by learning a subspace representation. The age of a test face is determined by the projection in the subspace that can best reconstruct the face image. However, their approach assumes all data has been labeled.

Most of the conventional methods for age estimation are intended for accurate estimation of the actual age. However, it is difficult to accurately estimate an actual age from a face image because age progression is person-specific and the aging subspace is obtained based on a incomplete database. For some applications, such

as digital signage, it is unnecessary to obtain the precise estimates of the actual age. Therefore, in this paper, we invoke the mechanism of human age perception, i.e. we limit the estimation to a few age ranges. We aim to use both labeled and unlabeled data. It is possible and easier for a user to label an age range of a person based on his/her face image, whereas it is very hard to label the actual age required in a actual age estimation system. This is the main reason why the existing methods, e.g. Geng et al.'s method, cannot use unlabeled data. To the best of our knowledge, we are the first to estimate age using both labeled and unlabeled face images.

Our contributions in this paper are two-fold. One is to propose the 2DLDA-FNN as a generic on-line or active learning paradigm and the second is to show that it can be another viable tool for active learning of facial age range classification.

The rest of this paper is organized as follows. The 2DLDA is introduced in section 2, An incremental version of the 2DLDA and active based on the 2DLDA are discussed in section 3. The experimental results are given in Section 4 and conclusion is presented in Section 5.

2 An Overview of 2DLDA

Suppose $\{(X_1^1, C_1), \dots, (X_{n_1}^1, C_1), \dots, (X_1^N, C_N), \dots, (X_{n_N}^N, C_N)\}$ are T image samples from N classes. $X_i^k \in R^{r \times w}$ ($r \times w$ image matrix) is the i^{th} sample of the k^{th} class C_k , for $i = 1, \dots, n_k$ and n_k is the number of samples in class C_k . Denote $\bar{X}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} X_i^k$ as the mean matrix of samples of the class C_k . Let $M = \sum_{k=1}^N \frac{n_k}{T} \bar{X}_k$ be the mean matrix of all samples. Several 2DLDA methods [15,34] have been proposed in face recognition. Without loss of generality, Kong's bilateral 2DLDA (B2DLDA) [10] is adopted in this paper as the 2DLDA to be extended to incremental 2DLDA. Concisely put, the B2DLDA is a general 2DLDA which finds a pair of discriminant vectors W_l and W_r satisfying:

$$\{W_l, W_r\} = \arg \max_{(W_l, W_r)} \frac{\sum_{k=1}^N \frac{n_k}{T} W_l^T (\bar{X}_k - M) W_r W_r^T (\bar{X}_k - M)^T W_l}{\frac{1}{T} \sum_{k=1}^N \sum_{i=1}^{n_k} W_l^T (X_i^k - \bar{X}_k) W_r W_r^T (X_i^k - \bar{X}_k)^T W_l} \quad (1)$$

The optimal W_l and W_r can be corresponding to the eigenvectors of $S_{wl}^{-1} S_{bl}$ and $S_{wr}^{-1} S_{br}$ respectively, where S_{wl} , S_{bl} are the left within-class and between-class scatter matrices of the training samples respectively; S_{wr} , S_{br} are the right within-class and between-class scatter matrices of the training samples respectively. The pseudo-code for the B2DLDA algorithm is given as follows.

Algorithm B2DLDA ($W_l, W_r, LB_1^1, LB_1^2, \dots, LB_1^{n_1}, \dots, LB_N^1, LB_N^2, \dots, LB_N^{n_N}, RB_1^1, RB_1^2, \dots, RB_1^{n_1}, \dots, RB_N^1, RB_N^2, \dots, RB_N^{n_N}$) = B2DLDA($X_1^1, X_2^1, \dots, X_{n_1}^1, \dots, X_{n_N}^N, m_l, m_r, N, n_1, n_2, \dots, n_N$)

Input: $X_1^1, X_2^1, \dots, X_{n_1}^1, \dots, X_{n_N}^N, m_l, m_r$. m_l and m_r are the number of the discriminant components of left and right B2DLDA transforms respectively.

Output: W_l, W_r, LB_i^j, RB_i^j . LB_i^j and RB_i^j are the reduced representations of X_i^j by W_l and W_r respectively.

1. Compute the mean, \bar{X}_i , of the i^{th} class, $i = 1, 2, \dots, N$
2. Compute the global mean, M .
3. Update S_{bl} and S_{wl}

$$S_{bl} = \sum_{i=1}^N n_i (\bar{X}_i - M)^T (\bar{X}_i - M) \quad (2)$$

$$S_{wl} = \sum_{i=1}^N \sum_{j=1}^{n_i} (X_j^i - \bar{X}_i)^T (X_j^i - \bar{X}_i) \quad (3)$$

4. Compute the first m_l eigenvectors, $\{\psi_i^L\}_{i=1}^{m_l}$, of $S_{wl}^{-1} S_{bl}$
5. $W_l \leftarrow [\psi_1^L, \psi_2^L, \dots, \psi_{m_l}^L]$
6. Update S_{br} and S_{wr}

$$S_{br} = \sum_{i=1}^N n_i (\bar{X}_i - M) (\bar{X}_i - M)^T \quad (4)$$

$$S_{wr} = \sum_{i=1}^N \sum_{j=1}^{n_i} (X_j^i - \bar{X}_i) (X_j^i - \bar{X}_i)^T \quad (5)$$

7. Compute the first m_r eigenvectors, $\{\psi_i^R\}_{i=1}^{m_r}$, of $S_{wr}^{-1} S_{br}$
8. $W_r \leftarrow [\psi_1^R, \psi_2^R, \dots, \psi_{m_r}^R]$
- 9.

$$LB_i^j = X_j^i * W_l, j = 1, 2, \dots, n_i, i = 1, \dots, N \quad (6)$$

$$RB_i^j = (X_j^i)^T * W_r, j = 1, 2, \dots, n_i, i = 1, \dots, N \quad (7)$$

3 Incremental 2DLDA and Active Learning

Inspired by the work on Incremental Linear Discriminant Analysis (ILDA) [9, 22, 30], we derive an exact solution of Incremental 2DLDA (I2DLDA) in this paper for updating the discriminant eigenspace where bursts of new class data are coming in sequentially. The idea is that the between-class and within-class matrices can be updated without much re-calculations. This extension of the 2DLDA is important because the I2DLDA inherits the advantages of the 2DLDA and the ILDA. Based on the I2DLDA, the small sample size problem [10] can be avoided as well, and it does not have to redo the entire training when a new sample is added. While our formulation provides an exact solution, the existing ILDA [9, 30] gives only approximate updates and thus it may suffer from numerical instability.

3.1 I2DLDA

Assume we are given t new samples and their labels, $Y = \{(Y_1, l_1), (Y_2, l_2), \dots, (Y_t, l_t)\}$. Without loss of generality, assume there are q_m new samples, which belong to the m^{th} class. The mean of the m^{th} class is updated as follows:

$$\bar{X}'_m = \frac{n_m \bar{X}_m + \sum_{Y_k \in Y \cap l_k = m} Y_k}{n_m + q_m} \quad (8)$$

$$n'_m = n_m + q_m \quad (9)$$

The updated overall mean is

$$M' = \frac{TM + \sum_{i=1}^t Y_i}{T + t} \quad (10)$$

The between-class scatter matrices are updated by

$$S'_{bl} = \sum_{c=1}^N n'_c (\overline{X}'_c - M')^T (\overline{X}'_c - M') \quad (11)$$

$$S'_{br} = \sum_{c=1}^N n'_c (\overline{X}'_c - M') (\overline{X}'_c - M')^T \quad (12)$$

where n'_c and \overline{X}'_c are the updated number of samples and mean of class C . The within-class scatter matrices are updated by

$$\begin{aligned} S'_{wl} = \sum_{c=1}^N \Sigma'_c = \sum_{c=1}^N \{ \Sigma_c + (\overline{Y}_c - \overline{X}_c)^T (\overline{Y}_c - \overline{X}_c) + \\ \frac{n_c^2}{(n_c + q_c)^2} \Sigma_{Y_k \in Y \cap l_k = c} (Y_k - \overline{X}_c)^T (Y_k - \overline{X}_c) + \\ \frac{q_c(q_c + 2n_c)}{(n_c + q_c)^2} \Sigma_{Y_k \in Y \cap l_k = c} (Y_k - \overline{Y}_c)^T (Y_k - \overline{Y}_c) \} \end{aligned} \quad (13)$$

$$\begin{aligned} S'_{wr} = \sum_{c=1}^N \Sigma'_c = \sum_{c=1}^N \{ \Sigma_c + \frac{n_c^2 q_c^2}{(n_c + q_c)^2} (\overline{Y}_c - \overline{X}_c) (\overline{Y}_c - \overline{X}_c)^T + \\ \frac{n_c^2}{(n_c + q_c)^2} \Sigma_{Y_k \in Y \cap l_k = c} (Y_k - \overline{X}_c) (Y_k - \overline{X}_c)^T + \\ \frac{q_c(q_c + 2n_c)}{(n_c + q_c)^2} \Sigma_{Y_k \in Y \cap l_k = c} (Y_k - \overline{Y}_c) (Y_k - \overline{Y}_c)^T \} \end{aligned} \quad (14)$$

where \overline{Y}_c is the mean of the new samples in Y belonging to the class c . If the samples belong to a new class, assume there are q_{N+1} new samples belong to the $(N+1)^{th}$ class, $n'_{N+1} = q_{N+1}$, the between class matrices are updated as

$$S'_{bl} = \Sigma n'_c (\overline{X}_c - M')^T (\overline{X}_c - M') \quad (15)$$

$$S'_{br} = \Sigma n'_c (\overline{X}_c - M') (\overline{X}_c - M')^T \quad (16)$$

The within-class matrix is updated

$$S'_{wl} = S_{wl} + \Sigma_{(N+1)l} \quad (17)$$

$$S'_{wr} = S_{wr} + \Sigma_{(N+1)r} \quad (18)$$

where $\Sigma_{(N+1)l}$ and $\Sigma_{(N+1)r}$ are the left and right covariance matrices of the $(N+1)^{th}$ class

$$\Sigma_{(N+1)l} = \Sigma_{Y_k \in Y \cap l_k = c} (Y_k - \overline{Y}_c)^T (Y_k - \overline{Y}_c) \quad (19)$$

$$\Sigma_{(N+1)r} = \Sigma_{Y_k \in Y \cap I_k = c} (Y_k - \bar{Y}_c)(Y_k - \bar{Y}_c)^T \quad (20)$$

Very recently, Li et al. [14] proposed an incremental 2DLDA based on a unilateral 2DLDA [15] which only does transform on one side of the image matrix, i.e., either left side or right side. Our method is different from [14] in that the 2DLDA adopted by our approach first extracts the 2D-LDA discriminative projections on both sides of the image matrix independently and then combine them through further processing. The principle motivation is to increase the information extracted from the covariance matrix. In order to do incremental learning, both ILDA and I2DLDA need to maintain one between-class covariance matrix and one within-class covariance matrix of every class. However, the size of the between-class covariance matrix and one within-class covariance matrix is much smaller than the ones of ILDA. Thus I2DLDA can overcome the limitation of the number of the class or chunk size in ILDA.

The experiments on a large database including labeled and unlabeled data show that active learning based on the I2DLDA approach trained in this manner can achieve results comparable or even outperform a framework trained in the conventional manner that requires much more labeling effort.

3.2 Pool-Based Active Learning

For many real-world learning problems including text classification and facial age estimation discussed in this paper, large collections of unlabeled data can be gathered at once. This motivates pool-based sampling [13]. In active learning, there is a pool of unlabeled points U and a (much smaller) pool of labeled points L . The goal is to iteratively pick the most informative points in U for labeling, obtain the labels from some oracle or teacher, add the labeled points to L , incrementally update the classifier using the newly added samples from U , and then iterate and see how fast the classifier converge to the final solution. An active learner has three components: (1) a classifier trained on the current set of labeled data; (2) a querying function that decides which instance in U to query at the next round and (3) an updated classifier after each query. We consider that a classifier is initially trained using a small number of randomly selected labeled examples called the seed set. Then repeat the process until either the evaluation rate reaches at a predefined value, or U is an empty set or until the oracle is no longer able to provide labels. During each round of active learning, n points are selected for labeling. We will refer to this as the batch size. The main difference between active learners is in the method to determine whether a point in U will yield valuable information if labeled.

Uncertainty sampling. The difference between an active learner and a passive learner is in the querying component, which brings us to the criterion for choosing the next unlabeled instance to query. All active learning scenarios involve evaluating the probability/believe of informativeness of unlabeled instances, which can either be generated de novo or sampled from a given distribution. There

have been many ways of formulating such query strategies described in the literature. A good survey can be found in [26]. For example, unlabeled examples to query are selected based on minimizing the version space within the SVM formulation [28]. Seung et al. [27] propose an algorithm called query by committee in which a committee of students is trained on the same data set and the next query is chosen according to the principle of maximal disagreement. In this paper, we use an uncertainty sampling approach [13] as a query strategy to perform active learning. Uncertainty sampling works by assigning an uncertainty score to each point in U and picking the n points with the highest uncertainty scores. These uncertainty scores are based on the predictions of the classifier currently trained on L . Uncertainty sampling method relies on probability estimates of class membership for all the examples in the active pool. Margin-based classifiers, for example SVM, has been used as a notion of uncertainty in previous work where class membership probabilities of the unlabeled examples are first estimated using the distance from the hyperplane for classifiers. The uncertainty score is inversely proportional to the absolute value of the distance from the present optimal hyperplane, where points closer to the hyperplane contain more uncertainty as to their class memberships. In this paper, we will compare our proposed approach with the SVM-based on entropy sampling [8] as state-of-the-art algorithm. LDA and its variants are feature extractors. It has to be coupled to a classifier, with the extracted feature vector serving as the input. Any classifier can be used. For simplicity, we choose the nearest neighbour classifier for investigation. How can the uncertainty be measured in the LDA domain? The output of the 2DLDA classifier is the distance of the query sample to the class instead of probabilities. For our 2DLDA, we will use the nearest neighbour classifier, which is one of the simplest classification schemes and can naturally handle multi-class problems. But, 2DLDA-nearest neighbour does not admit a natural notion of uncertainty in multi-class classification, and hence, it is unclear how to estimate the probability of misclassification for a given data point.

Data selection. We propose to select the unlabeled data that is the furthest nearest neighbour to the 2DLDA classifier among all the nearest neighbours for all the classes. We call this sample selection method as FNN (Furthest Nearest Neighbour). This means all the unlabelled or uncertain data is tested and for each, compute its nearest neighbour in the 2DLDA projection subspace as before. Choose the one which has the furthest nearest neighbour. The FNN is heuristic but we offer the following rationale.

Rationale: The data with the furthest nearest neighbour distance is deemed to have the highest probability of uncertainty.

If the nearest neighbour turns out to be incorrectly classified, this will imply a very drastic step forward in learning. If we assume that the nearer the example is to a data the higher the probability that the example is classified correctly, then one that is furthest away will have the least probability of being correctly classified. On the other hand, if the selected data turns out to be correctly classified, then it provides the highest generalization learning among

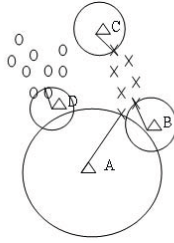


Fig. 1. An example for showing the FNN data selection

all other unlabeled data. Either way, the learning will, with high probability, be fast. So we want to learn from this example.

The computation of the FNN is explained thus. Assume the projection matrices of the bilateral 2DLDA to W_l and W_r respectively, and the vector of an image, X , in the subspace is $W_l X W_r$. The data in the active pool U will be selected next according to the FNN criterion as follows :

$$\max_i \left\{ \min_{X \in L} \|W_l Z_i W_r - W_l X W_r\|, i = 1, 2, \dots, u \right\} \quad (21)$$

where L represents the current training set, assuming there are u samples in U , represented as $Z_i, i = 1, 2, \dots, u$. An example of the FNN is shown in Fig. 1. Assume we have two classes of samples, marked as "o" and "x" and their 2D feature space distributions are shown in Fig. 1. Symbol "Δ" represents the four unlabeled samples been projected to this subspace. The nearest neighbours of the four unlabeled samples are shown connected with them respectively. "A" is the first sample to be selected by the FNN selection method because it is the furthest nearest neighbour. Furthermore, our method does not make any assumption about the number of the new class to be labeled and allowing application to huge datasets with a large number of categories.

The new classifier is incrementally learned using the added samples, and uncertainty scores are produced for the unlabeled data in the pool. The learning process is shown in Fig. 2. The threshold in Fig. 2 is the value of the accuracy that the user expected. It can be set by the user depend on the application. In the following experiments, we set the threshold to be 1, i.e. active learning continues till the active pool is exhausted. It should be noted that the algorithm can select more than one sample at each iteration and this make it possible to incremental update classifier by processing selected samples in one batch. This is very important to speed up the active learning to reach an expected accuracy.

In order to remove possible outliers, we introduce a criterion to reject a furthest NN if it exceeded a certain threshold distance. Thus it will not affect the active learning significantly. The threshold is determined empirically from experiments. In our experiments, we found that the case of outliers is not serious. This can be observed by the variance of the furthest nearest neighbour distances for active learning rounds which are found to be small. Hence, the sensitivity of the performance on the threshold is minor. In principle though, note that by the very nature of the nearest neighbour rule, outliers will not be selected.

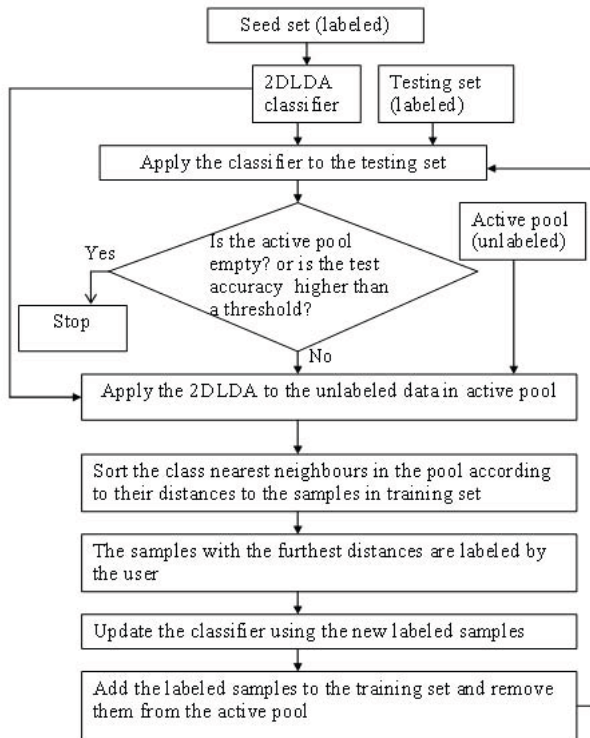


Fig. 2. The flowchart of the proposed active learning

4 Experimental Results

There are few publicly available age databases. The FG-NET [3] and Morph [25] databases have been made available for research in areas related to age-progression. The FG-NET database contains 1,002 face images from 82 subjects, with approximately 10 images per subject. In the MORPH database, there are 1,724 face images from 515 subjects. Each subject has around 3 aging images. We use these two databases (a total of 2726 images) together with an unlabeled data set collected by us to evaluate the performance of the proposed method. For the unlabeled face data, we collected a database which consists of three sources: (1) collected in our lab and Singapore Science Centre; (2) frontal face images in public databases including LFW (Labeled Faces in the Wild), PEAL (Pose, Expression, Accessories, and Lighting) and PIE (Pose, Illumination, and Expression); (3) collected from internet. There are a total of 4000 unlabeled face images of 1713 persons. The Viola-Jones face detector [21,31] was used to detect faces and all detected faces are then geometrically normalized to 88×88 images. Some labeled samples can be seen in Fig 3 and some unlabeled samples can be seen in Fig. 4.



Fig. 3. Some labeled samples. From first row to the fourth row: child, teen, adult and senior adult.



Fig. 4. Some unlabeled samples

In this paper, we are interested in the perceived age range instead of actual age. For our study, we define four age groups: child, teen, adult and senior adult. The age ranges of the four groups are 0-11, 12-21, 22-60 and 61 and above respectively. Our focus is only investigating our proposed method and the partitioning adopted is only for proof of principle. Nevertheless, this partition is not without basis. In general, a person's appearance will undergo noticeable changes due to physiological and social factors. The first is at puberty, for girls it is around the age of twelve. The average graduation age of the undergraduate study and find the first job is about 21. The retirement age of the male people in the world is about 60. If different age groups (slight different on the boundary of age ranges) are used, the results should not be affected seriously because in general there is no large variance of the face appearance within each of the ranges defined in this paper. E.g. the large variance could happen from 0-11. The user is queried about the age range of the selected unlabeled data and labels it to one of four age groups defined in this paper. It is very hard for the user to label the actual age by observing a face image. So our method can not predict the exact age given the face unless we are provided with labeled age data, i.e. it is true that the method proposed in this paper can predict actual age if the user can label the selected data with actual age. However, there is greater likelihood to wrongly label actual age of given image respectively for a large dataset. We can predicate more age group by dividing the age range into more groups. However, the accuracy depends on the ability of the human to estimate finer age groups.

An initial classifier is trained using a seed set which is composed of randomly selecting half of the subjects of the FG-NET and Morph database respectively. The remaining is used as the testing set. For each round of the active learning, the samples in the unlabeled samples pool are sorted by the FNN and then 5 samples which are at the top of the pool are selected and labeled by the user. They are then added to the training set and the 2DLDA classifier is updated using the newly added samples. It should be noted that the test data does not

contain the images of persons also used for training. This avoids the inadvertent learning of face recognition instead. Two-fold cross-validations have been conducted and the average classification accuracies versus the number of the learning rounds are shown in Fig. 5. One interesting aspect of the results, particularly on the data set, is that the error rate can be much lower when only a subset of U has been labeled as opposed to when all of U being labeled. This phenomenon has been observed in other works on active learning as well (e.g., [19]). Stopping the labeling process early can therefore be very useful in reducing overall classification error. One possible reason is that stopping the process early helps to avoid over-fitting. The performance of the proposed FNN active learning is

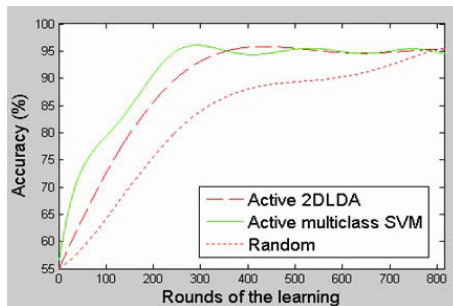


Fig. 5. Performance comparison of random selection (bottom), FNN active learning (middle) and active SVM. Five samples are selected at each learning round.

compared with the baseline random sampling method and the state-of-the-art method: active multi-class Support Vector Machines (SVM) [8] that are based on the one-vs-one formulation of binary classifiers (a classifier trained for each pair of classes) to handle multi-class problems. Probability estimates for the multi-class case are obtained through pairwise coupling [33] and binary probabilities needed are estimated follow Platt method [17]. The results are shown in Fig. 5. In our approach, we found that the method converges quickly, thus verifying our proposed criterion for the unlabeled data with the highest probability of uncertainty. It has comparable accuracy with active multiclass SVM. The convergence rate is faster for SVM i.e. it needs less selected examples for labeling. However, the active 2DLDA is much faster in terms of generating the training results. In our implementation of the active multiclass SVM, LIBSVM toolbox [16] is adopted that implements the uncertainty sampling and probability estimation in the multiclass problem mentioned above. "Linear" kernel is adopted. The comparison of the training time is given in Table 1. The Acer Veriton 7900 C2D 2.66Ghz 4GB/320GB personal computer is used and the speed for training the active 2DLDA is found to be about 8 times faster than the active SVM.

We quantify the reduction in the number of training examples required for the FNN to obtain similar accuracy as random selection. In Fig. 5, for each round of active learning, the number of rounds to achieve similar accuracy by fixing a value at Y-axis. The results, tabulated in Table 2, show that FNN selection

Table 1. The comparison of the training time between the active 2DLDA and active multiclass SVM

Method	Active SVM	Active 2DLDA	Random
Training time (s)	2267	283	221

Table 2. Reduction of the number of the learning rounds needed for the active 2DLDA (A-2DLDA) and active SVM (A-SVM) for getting the similar accuracy with random selection. The number needed and corresponding accuracy are represented as "number(accuracy)" in the first three columns

Random	A-SVM	A-2DLDA	reduction(A-SVM)	reduction(A-2DLDA)
95 (62.5%)	10 (63.1%)	30 (62.7%)	89.47%	68.42%
115 (65.7%)	25 (66.2%)	50 (66%)	78.26%	56.52%
160 (71.2%)	30 (70.8%)	80 (71%)	81.25%	50.00%
370 (86.3%)	170 (86.6%)	220 (86%)	54.05%	40.54%
790 (94.7%)	248 (95.2%)	380 (95%)	68.61%	51.90%
			74.33(average)	53.48% (average)

can obtain similar accuracy with random selection but uses about 55% fewer samples.

Based on the proposed active learning approach, an age classification prototype has been developed. The age ranges of the subjects can be simultaneously estimated automatically. We used the Logitech Webcam Pro 9000 to capture face images while face detection and tracking are done using the face detector of the OpenCV 2.0 library [21] and the kernel-based mean shift algorithm [1] respectively. Future research includes the robustness to expression, pose and lighting variation.

5 Conclusion

In this paper, an active learning approach has been proposed to classify a face image to pre-defined age category. In order to solve the incomplete data problem in facial age estimation, both labeled and unlabeled data have been used. The proposed methods that combine active learning with a solution to the age estimation provide a good trade off between achieving a low error rate and reducing data labeling cost better than the random baseline. Instead of using a randomly selected training set, the learner has access to a pool of unlabeled instances and can request the labels for some number of them using a new data informative measure called FNN. An incremental 2DLDA is proposed to update the discriminant subspace instead of full re-training whenever a new training sample is added. A closed-form solution for updating the between-class scatter matrix and within-class scatter matrix using the new samples is derived. Empirical experiments on FG-NET, Morph data and a large unlabeled face database collected by the authors for age classification problems show that approach can achieve

results much faster than random selection and get the similar result with random selection using about 55% fewer samples. It can achieve the comparable results with active SVM but is much faster than active SVM in terms of generating the training results.

In the future, we could relax the condition of applying 2DLDA to use LDA also, then we can apply the FNN on other 1D databases, e.g. UCI data sets [29].

References

1. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25, 564–577 (2003)
2. Dasgupta, S.: Corse sample complexity bounds for active learning. In: *Proceedings of Neural Information Processing Systems (NIPS)* (2006)
3. The FG-NET Aging Database (2010), <http://www.fgnet.rsunit.com>
4. Freund, Y., Seung, S., Shamir, E., Tishby, N.: Selective sampling using the query by committee algorithm. *Machine Learning Journal* 28, 133–168 (1997)
5. Fu, Y., Huang, T.S.: Human age estimation with regression on discriminative aging manifold. *IEEE Transactions on Multimedia* 10, 578–584 (2008)
6. Geng, X., Zhou, Z.-H., Zhang, Y., Li, G., Dai, H.: Learning from facial aging patterns for automatic age estimation. In: *Proceedings of ACM Conference on Multimedia*, pp. 307–316 (2006)
7. Jain, P., Kapoor, A.: Active Learning for Large Multi-class Problems. In: *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1–8 (2009)
8. Joshi, A.J., Porikli, F., Papanikolopoulos, N.: Multi-Class Active Learning for Image Classification. In: *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 2372–2379 (2009)
9. Kim, T.K., Wong, S.-F., Stenger, B., Kittler, J., Cipolla, R.: Incremental linear discriminant analysis using sufficient spanning set approximations. In: *Proceedings of IEEE International Conference on Pattern Recognition*, pp. 1–8 (2007)
10. Kong, H., Wang, L., Teoh, E.K., Wang, J.-G., Venkateswarlu, R.: A framework of 2D Fisher Discriminant Analysis: application to face recognition with small number of training samples. In: *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1083–1088 (2005)
11. Kwon, Y.H., Lobo, N.D.V.: Age Classification from Facial Images. *Computer Vision and Image Understanding* 74, 1–21 (1999)
12. Lanitis, A., Draganova, C., Christodoulou, C.: Comparing Different Classifiers for Automatic Age Estimation. *IEEE Trans. on SMC-Part B, Cybernetics* 34, 621–628 (2004)
13. Lewis, D.D., Gale, W.A.: A sequential algorithm for training text classifiers. In: *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, vol. 34, pp. 3–12 (1994)
14. Li, G., Liang, D., Huang, Q., Jiang, S., Gao, W.: Object tracking using incremental 2D-LDA learning and Bayes inference. In: *Proceedings of IEEE International Conference on Image Processing*, pp. 1567–1571 (2008)
15. Li, M., Yuan, B.: 2D-LDA: a novel statistical linear discriminant analysis for image matrix. *Pattern Recognition Letters* 26, 527–532 (2002)
16. LIBSVM toolbox (2010), <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

17. Lin, H.-T., Lin, C.-J., Weng, R.: A note on Platt's probabilistic outputs for support vector machines. *Machine Learning* 68, 267–276 (2007)
18. Liu, A., Jun, G., Ghosh, J.: Spatially Cost-Sensitive Active Learning. In: *Proceedings of SIAM International Conference on Data Mining*, pp. 814–825 (2008)
19. Liu, A., Jun, G., Ghosh, J.: Active learning of hyperspectral data with spatially dependent label acquisition costs. In: *Proceedings of IEEE International Geoscience and Remote Sensing Symposium*, pp. V-256–V-259 (2009)
20. Nguyen, T.T., Binh, N.D., Bischof, H.: Efficient boosting-based active learning for specific object detection problems. *International Journal of Electrical, Computer, and Systems Engineering* 3, 150–155 (2009)
21. OpenCV (2010), <http://opencv.willowgarage.com/wiki>
22. Pang, S., Ozawa, S., Kasabov, N.: Incremental Linear Discriminant Analysis for Classification of Data Streams. *IEEE Trans. on SMC-Part B* 35, 905–914 (2005)
23. Ramanathan, N., Chellappa, R.: Modeling age progression in young faces. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 387–394 (2006)
24. Ramanathan, N., Chellappa, R., Biswas, S.: Computational methods for modeling facial aging: A survey. *Journal of Visual Languages and Computing* 20, 131–144 (2009)
25. Ricanek Jr., K., Tesafaye, T.: MORPH: a longitudinal image database of normal adult age-progression. In: *Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition*, pp. 341–345 (2006)
26. Settles, B.: Active learning literature survey. *Computer Sciences Technical Report 1648*, University of Wisconsin-Madison (2009)
27. Seung, H.S., Opper, M., Sompolinsky, H.: Query by committee. In: *Proceedings of Computational Learning Theory*, pp. 287–294 (1992)
28. Tong, S., Koller, D.: Support Vector Machine Active Learning with Applications to Text Classification. *Journal of Machine Learning Research* 2, 45–66 (2002)
29. UCI Machine Learning Repository : Data Sets 12 (2010), <http://archive.ics.uci.edu/ml/datasets.html>
30. Uray, M., Skocaj, D., Roth, P.M., Bischof, H., Leonardis, A.: Incremental LDA Learning by Combining Reconstructive and Discriminative Approaches. In: *Proceedings of British Machine Vision Conference*, pp. 272–281 (2007)
31. Viola, P., Jones, M.J.: Robust Real-Time Face Detection. *International Journal of Computer Vision* 57, 137–154 (2004)
32. Wang, J.-G., Yau, W.-Y., Wang, H.L.: Age Categorization via ECOC with Fused Gabor and LBP Features. In: *Proceedings of IEEE Workshop on Applications of Computer Vision*, pp. 313–318 (2009)
33. Wu, T.-F., Lin, C.-J., Weng, R.C.: Probability Estimates for Multi-class Classification by Pairwise Coupling. *Machine Learning Research* 5, 973–1005 (2004)
34. Ye, J., Janardan, R., Li, Q.: Two dimensional linear discriminant analysis. In: *Proceedings of Neural Information Processing Systems, NIPS* (2004)

Real-Time Human Detection Using Relational Depth Similarity Features

Sho Ikemura and Hironobu Fujiyoshi

Dept. of Computer Science, Chubu University
Matsumoto 1200, Kasugai, Aichi, 487-8501 Japan
si@vision.cs.chubu.ac.jp, hf@cs.chubu.ac.jp
<http://www.vision.cs.chubu.ac.jp>

Abstract. Many conventional human detection methods use features based on gradients, such as histograms of oriented gradients (HOG), but human occlusions and complex backgrounds make accurate human detection difficult. Furthermore, real-time processing also presents problems because the use of raster scanning while varying the window scale comes at a high computational cost. To overcome these problems, we propose a method for detecting humans by Relational Depth Similarity Features (RDSF) based on depth information obtained from a TOF camera. Our method calculates the features derived from a similarity of depth histograms that represent the relationship between two local regions. During the process of detection, by using raster scanning in a 3D space, a considerable increase in speed is achieved. In addition, we perform highly accurate classification by considering of occlusion regions. Our method achieved a detection rate of 95.3% with a false positive rate of 1.0%. It also had a 11.5% higher performance than the conventional method, and our detection system can run in real-time (10 fps).

1 Introduction

There has recently been interest into the implementation of techniques that will assist in comprehending the intentions of people within spaces such as offices, homes, and public facilities. In order to implement techniques of monitoring people in this manner, it is necessary to know where people are within such a space, and it has become a challenge to implement human detection that is highly accurate and also fast. There has been much research in the past into human detection, and various different methods have been proposed [1] [2] [3] [4] [5]. Among human detection methods that use conventional visible-light cameras, there are methods that involve statistical training with local features and boosting. Gradient-based features such as HOG [1], EOH [2], and edgelets [5] are often used as local features, and there have been reports that these enable the implementation of highly accurate human detection. However, gradient-based features obtained from visible-light camera images encounter difficulties in perceiving the shapes of human beings when there are complex backgrounds and when people overlap each other, and the detection accuracy can drop as a result.

To counter this problem, Ess et al. have proposed a highly-accurate human detection method for confusing scenes, using depth information obtained by stereo cameras [6]. However, the acquisition of depth information by stereo cameras necessitates correspondence calculations between images by camera calibration and stereo matching, so the processing costs are high and real-time detection is difficult. In addition, since the sizes of the humans within the image is unknown, conventional human detection methods also have problems in that repeated raster scans while varying the scale of the detection window increases the computational cost and makes real-time processing difficult.

This paper proposes a real-time human detection method that uses depth information obtained from a time-of-flight (TOF) camera and can cope with overlapping people and complex scenes. The proposed method uses depth information obtained from the TOF camera to calculate relational depth similarity features (RDSFs) that determine depth information for local regions, and constructs a final classifier by Real AdaBoost. It uses the thus-constructed classifiers to detect humans, and implements faster raster scanning of detection windows in a 3D space and also improves the detection accuracy by considering occlusion regions.

2 Local Features Based on Depth Information

A TOF camera is a camera that measures the distance to an object by measuring the time taken for infrared light that is emitted from LEDs located around the camera to be reflected by the object being detected and observed by the camera. In this study, we use a MESA SR-3100 as the TOF camera. The SR-3100 can acquire depth information in real-time from 0.3 m to 7.5 m (with a resolution of 22 mm at 3 m), but it cannot photograph outdoors so it is limited to use indoors. When detecting humans, it is considered effective to use depth information obtained by a TOF camera to perceive the depthwise relationships of human bodies and the background. Thus this method proposes the use of a relational depth similarity feature obtained from depth distributions of two local regions.

2.1 Relational Depth Similarity Features

A relational depth similarity feature is used to denote the degree of similarity of depth histograms obtained from two local regions. As shown in Fig. 1, we divide each depth image into local regions that are cells of 8 x 8 pixels, and select two cells. We compute depth histograms from the depth information of each of the two cells selected in this way, then normalize them so that the total value of each depth histogram is 1. If each bin of the two normalized depth histograms p and q created from the thus computed m bins is p_n and q_n , we compute the degree of similarity S between them from the Bhattacharyya distance [7] and use that as an RDSF.

$$S = \sum_{n=1}^m \sqrt{p_n q_n} \quad (1)$$

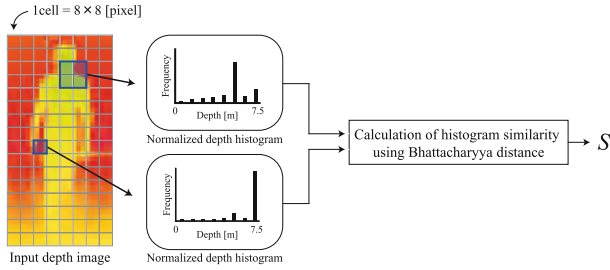


Fig. 1. Calculation of RDSF

Since the RDSF is a feature obtained from the degree of similarity of depth information for two regions, it becomes a feature that expresses the relative depth relationship between the two regions.

2.2 Varied Rectangular Region Sizes

Based on the processing of Section 2.1, we calculate an feature vector of RDSF by calculating the degrees of similarity for all combinations of rectangular regions, as shown in Fig. 2. During this process, we use Equation (2) for normalization. In this case, if p_n is the n th bin of the depth histogram, the n th bin p'_n of the normalized depth histogram can be obtained from the following equation:

$$p'_n = \frac{p_n}{\sum_{i=1}^m p_i} \quad (2)$$

With this method, the detection window size is set to 64 x 128 pixels so it can be divided into 8 x 16 cells. There are 492 rectangular regions obtained by varying the cell units of the rectangular region from 1 x 1 to 8 x 8 to compute depth histogram. To calculate the RDSF from combinations of the 492 rectangular regions, ${}_{492}C_2 = 120,786$ feature candidates are extracted from within one detection window.

2.3 Faster Depth Histogram Calculations by Integral Histograms

To reduce computational costs during the feature calculations, this method uses integral histograms [8] to compute the depth histograms rapidly. We first quantize the depth of each pixel to a 0.3-m spacing. Since this study divides the distances 0 m to 7.5 m by a 0.3-m spacing, that means we compute depth histograms formed of 25 bins. We then create 25 quantized images in corresponding to bin n , as shown in Fig. 3, and compute an integrated image $ii^n(u, v)$ from the quantized images $i^n(u, v)$, using Equations (3) and (4):

$$s^n(u, v) = s^n(u, v - 1) + i^n(u, v) \quad (3)$$

$$ii^n(u, v) = ii^n(u - 1, v) + s^n(u, v) \quad (4)$$

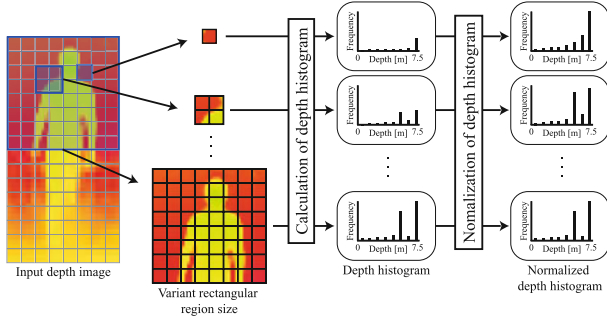


Fig. 2. Normalization of depth histograms for various rectangular region sizes

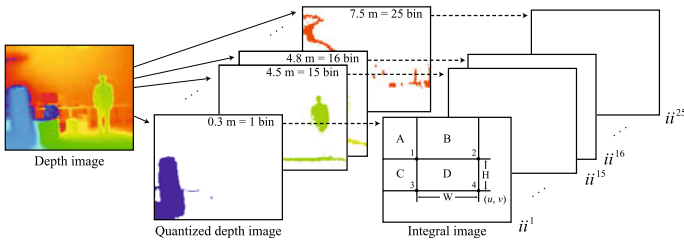


Fig. 3. Calculation of integral histogram

In this case, $s^n(u, v)$ denotes the sum of pixels in the rows of bin n and $ii^n(u, v)$ denotes the sum of s^n of the columns. Note that we assume that $s^n(u, -1) = 0$ and $ii^n(-1, v) = 0$. In the calculation of the n th bin D^n of the depth histogram from the region D in Fig. 3, it would be sufficient to obtain the sum from four points of the n th integrated image ii^n , from the following equation:

$$D^n = (ii^n(u, v) + ii^n(u - W, v - H)) - (ii^n(u - W, v) + ii^n(u, v - H)) \tag{5}$$

This makes it possible to rapidly obtain the value of the n th bin of the depth histogram, irrespective of the size of the region.

3 Human Detection Using Depth Information

The flow of human detection in accordance with the proposed method is shown in Fig. 4. The proposed method first performs a raster scan of the detection windows in a 3D space. It then computes the RDSFs from the detection windows. It judges whether there are occlusions in the calculated features, and uses Real AdaBoost to classify whether each detection window is of a human or a non-human object. Finally, it integrates the detection windows that have been classified as human by mean-shift clustering in the 3D space, to determine the location of human.

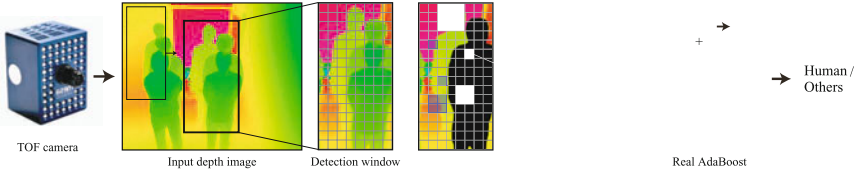


Fig. 4. Flow of human detection using depth information

3.1 Construction of Classifiers by Real Adaboost

The proposed method uses Real AdaBoost [9] in the human classification. Real AdaBoost obtains degrees of separation from the probability density functions for each dimension of features in positive classes and negative classes, and selects the features that enable the greatest separation between positive and negative classes as weak classifiers. Since the degrees of separation are handled as evaluated values during this process, the output of the classification results can be done as real numbers. If a weak classifier selected by the training is $h_t(x)$, the final classifier $H(x)$ that is constructed is given by the following equation:

$$H(x) = \text{sign}\left(\sum_{t=1}^T h_t(x)\right) \quad (6)$$

3.2 Raster Scanning in 3D Space

Conventional human detection methods involve repeated raster scans while the scale of the detection window is varied, so there are many detection windows that do not match the dimensions of humans. With the proposed method, we determine the detection windows to correspond to the sizes of humans, by using depth information to perform raster scans in a 3D space, which speeds up the processing. With $y_w = 0$, 60×180 [cm] detection windows in the $x_w - z_w$ plane are subjected to raster scanning, as shown in Fig. 5. The 3D coordinates of each detection window obtained by the raster scan in the 3D space are projected onto image coordinates $[u, v]^T$, using Equation (7), and a feature is calculated from the depth information within the region corresponding to the projected coordinate position.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (7)$$

$$\mathbf{P} = \mathbf{A}[\mathbf{R}|\mathbf{T}] \quad (8)$$

The matrix \mathbf{P} is a perspective projection matrix which is computed from an intrinsic parameter obtained by camera calibration, a rotation matrix \mathbf{R} that

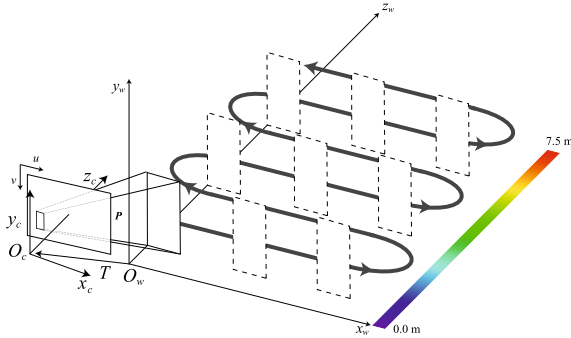


Fig. 5. Raster scanning in 3D space

is an extrinsic parameter, and a translation vector \mathbf{T} . Simple camera calibration can be done to enable the TOF camera to acquire the global coordinates (x_w, y_w, z_w) within a $5 \times 4 \times 7.5$ [m] space corresponding to the image coordinates (u, v) . In this case, the TOF camera uses a CCD sensor to observe infrared light, so the intrinsic parameter \mathbf{A} is similar to that of an ordinary camera. In addition, Mure-Dubois, et al. have compared published intrinsic parameters and the results of actual calibrations and confirmed that the intrinsic parameters are close [10].

3.3 Classification Adjusted for Occlusions

In a confusing scene in which a number of people are overlapping, occlusions can occur in the human regions that are being observed. Depth information extracted from an occlusion region is the cause of the output of erroneous responds for weak classifiers. Thus we ensure that any output of weak classifiers that perceive such occlusion regions is not integrated into the final classifier without modification. Since the proposed method performs a raster scan of a detection window in a 3D space, the position of the window in global coordinates is known. In this case, we determine that any object region that is closer to the camera than the detection window is an occlusion, by comparing depth information acquired from the TOF camera, and use that in the classification. In this study, we assume that when there is natural overlapping between people, the depth from one person who is in the front and another person who is in the rear will be at least 0.3 m, and that any object region that is at least 0.3 m closer than the detection window for a person being detected is an occlusion.

Extraction of occlusion regions. We use the depth z_w of the detection window during the raster scanning of the 3D space to determine the threshold for occlusion judgement. If we assume that each pixel in the detection window is (u, v) and the depth map thereof is $d(u, v)$, the occlusion label $O(u, v)$ at each set of coordinates is given by the following equation:

$$O(u, v) = \begin{cases} 1 & \text{if } d(u, v) < z_w - 0.3 \text{ m} \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

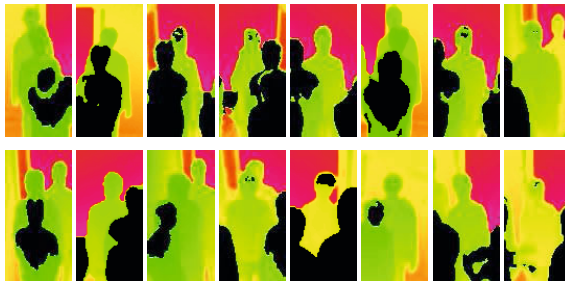


Fig. 6. Examples of occluded regions

The extracted occlusion regions are shown in Fig. 6 as black areas.

Classification from consideration of occlusion regions. If we assume that the proportion of occlusion regions existing within a rectangular region B_t , which is the target of the t th weak classifier $h_t(x)$, is an occlusion rate OR_t , it can be obtained from the following equation:

$$OR_t = \frac{1}{B_t} \sum_{(u,v) \in B_t} O(u,v) \quad (10)$$

Using the thus-computed occlusion rate OR_t , the final classifier $H'(x)$ from consideration of occlusion regions is expressed by Equation (11). During this time, the output of weak classifiers that have been computed from regions in which occlusions occur can be restrained by applying the proportion of occlusions as weighting to the weak classifiers.

$$H'(x) = \text{sign}\left(\sum_{t=1}^T h_t(x) \cdot (1 - OR_t)\right) \quad (11)$$

If the classification by final classifiers is done without considering occlusion regions, as shown in Fig. 7(a), the output of a large number of weak classifiers is a disadvantage and as a result, non-human objects are mistakenly classified. On the other hand, Fig. 7(b) shows an example in which the output of final classifiers is obtained from consideration of occlusion rates, in which humans are classified correctly.

3.4 Mean-Shift Clustering in 3D Space

In conventional human detection with a visible-light camera [3], the detection windows that have been classified as representing humans are integrated by mean-shift clustering [11] and taken as detection results. However, with mean-shift clustering alone in an image space, detection windows could be erroneously integrated if humans overlap in them, as shown in (b) and (d) of Fig. 8. With the proposed method, mean-shift clustering is done within a 3D space, as shown

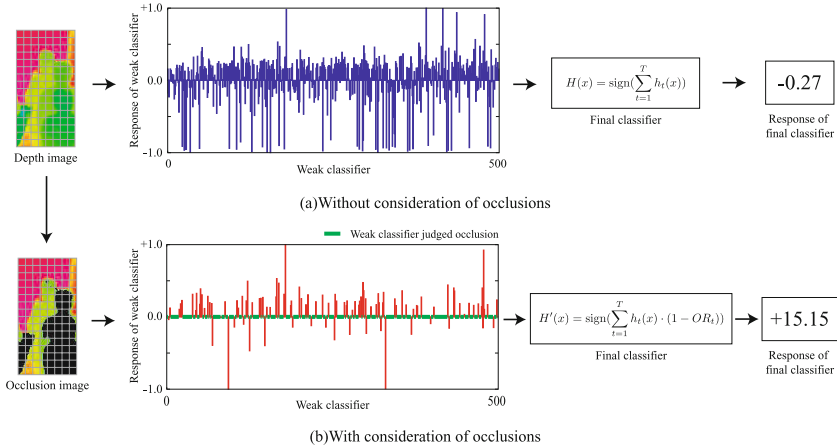


Fig. 7. Examples of classifications with and without consideration of occlusions

in (c) and (e) of Fig. 8. Since this enables the separation of clusters by depth information, even when humans are overlapping, the erroneous integration of detection windows can be suppressed.

3D mean-shift clustering calculates the mean-shift vector $m(\mathbf{x})$ from Equation (12). In this case, \mathbf{x} denotes the center coordinate of the detection window and \mathbf{x}_i denotes the 3D coordinate of each data item. k is a kernel function and h is the bandwidth, which in this study is taken to be $h = 0.3$ m.

$$m(\mathbf{x}) = \frac{\sum_{i=1}^n \mathbf{x}_i k\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^n k\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x} \quad (12)$$

4 Evaluation of the Proposed Method by Classification Experiments

We performed evaluation experiments to confirm the validity of the proposed method.

4.1 Database

For the database, we used sequences shot by a TOF camera. We installed the TOF camera at a height of approximately 2.5 m indoors, and targeted scenes of people walking and scenes in which a number of people overlap. We used 1346 positive examples for training and 10,000 negative examples for training, taken from sequences that had been shot indoors. In the evaluation, we used 2206 positive samples for evaluation and 8100 negative samples for evaluation. Since

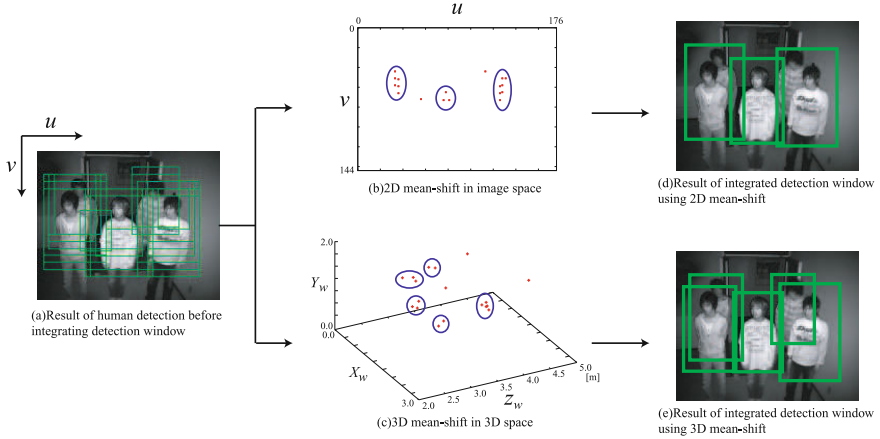


Fig. 8. Integration of detection windows by using mean-shift clustering

the TOF camera was set up to take pictures up to a maximum distance of 7.5 m indoors, it was difficult to use it to photograph the whole bodies of a number of humans. For that reason, the top 60% of the whole bodies of the humans were the targets for these experiments. Part of the database that was used for evaluation is shown in Fig. 9.

4.2 Feature Evaluation Experiments

Using the database for evaluation, we performed human classification experiments and compared them by feature classification accuracy. In the experiments, we compared features by using HOG features extracted from depth images, RDSFs, and both HOG features and RDSFs. Note that since these experiments were intended to evaluate features, there was no classifications adjusted for occlusions. We use receiver operating characteristic (ROC) curves in the comparison of the experiment results. A ROC curve plots false positive rate along the horizontal axis

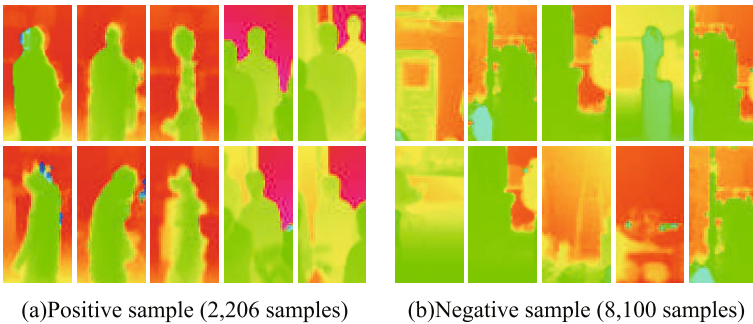


Fig. 9. Examples of test data

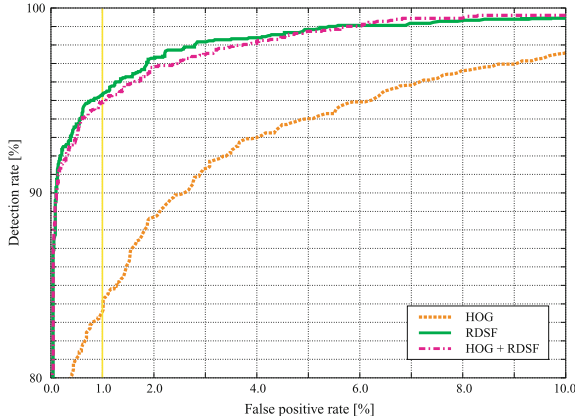
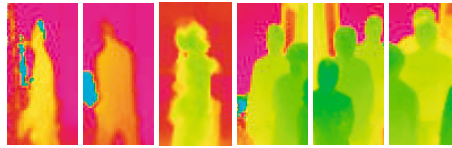


Fig. 10. Results of using features



(a) Example of missed detection during classification using HOG features



(b) Example of missed detection during classification using depth histogram features (variable of rectangular size)

Fig. 11. Examples of missed detection during classification

and detection rate along the vertical axis. It is possible to compare detection rate with respect to false positive rate by varying the classifier thresholds. The detection performance is better towards the top left of the graph.

The results of feature evaluation experiments are shown in Fig. 10. RDSFs gave a detection rate of 95.3% with a false positive rate of 1.0%, which is an improvement of 11.5% over the classification rate of HOG features of depth images. A comparison of RDSFs alone and features obtained by both HOG features and RDSFs showed that the detection accuracy was the same. This shows that RDSFs are mainly (98%) selected during the weak classifier training and HOG features do not contribute to the classification. Examples of missed classifications are shown in Fig. 11. It is clear that the samples that tended to be erroneously classified involved large variations in shape or occlusions.

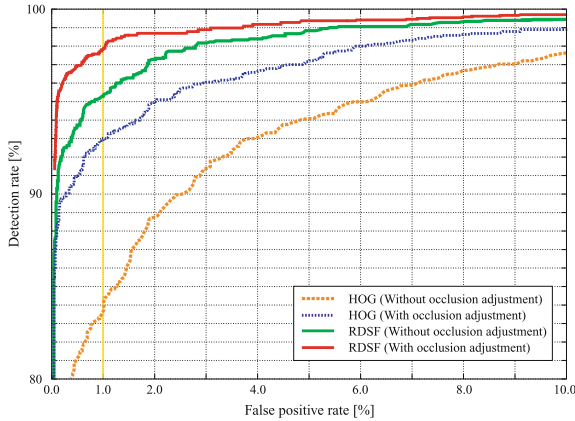


Fig. 12. Results of occlusion consideration

4.3 Evaluation Experiments Adjusted for Occlusions

To demonstrate the validity of occlusion adjustment in the proposed method, we evaluated it by human classification experiments.

The results of evaluation experiments with and without occlusion adjustment are shown in Fig. 12. RDSFs with occlusion adjustment gave a detection rate of 97.8% with a false positive rate of 1.0%, which makes it clear that this method enables highly accurate classification even when occlusions occur. In addition, the detection rate improved even with HOG features with occlusion adjustment. This shows that it is possible to suppress the effects of occlusion regions by using occlusion rates to weight weak classifiers that are valid for the classification, to obtain output of the final classifiers.

4.4 Features Selected by Training

Features that weak classifiers have selected in the Real AdaBoost training are shown in Fig. 13. With the HOG features of (a), features are selected in such a manner that the boundary lines between humans and the background such as the edges of the head and shoulders are perceived. It is clear that features of the upper half of bodies with few shape variations are often selected, as the tendency of training of up to 50 rounds. Next, with the RDSFs of (b), the selection is such that combinations of adjoining human regions and background regions are perceived in the first and third training rounds. Differing from the perception of boundaries at lines, such as with HOG features, boundaries are perceived by regions with RDSFs. This is considered to make the method robust in positioning humans. Boundaries were also perceived in the ninth and eleventh training rounds, but separated rectangular regions were selected, which differs from the first and third rounds. This is considered to make it possible to absorb variations in boundary position, since there are large variations in the lower halves of human bodies. In each of the second, fifth, and seventh training rounds,

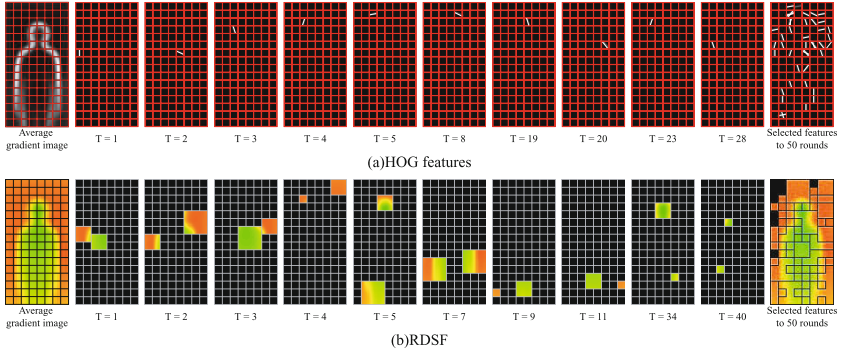


Fig. 13. Features selected by learning

Table 1. Computational costs per frame [ms]

	Processing cost of 1 detection window	Total (361 windows)
Feature calculation	0.067	24.31
Classification	0.125	45.34
Integration of windows	–	31.97
Total	–	101.62

regions that tend to determine vertical or lateral symmetrical depth relationships of the human shape were selected. In each of the thirty-fourth and fortieth training rounds, two regions in the interior of the human were selected. When there are rectangular regions of the same depth, those two rectangular regions can be taken to represent a surface of human. The tendency with up to 50 rounds of training makes it clear that large rectangular regions were selected during the initial training rounds to determine the depth relationships of large-scale human regions. As the training proceeded, the selection was such that local depth relationships were perceived by selecting smaller rectangular regions.

4.5 Real-Time Human Detection

The examples of human detection using raster scanning of detection windows in a 3D space are shown in Fig. 14. From (a), we see that an object of a similar height to people is not detected erroneously and only the people are detected. From (b) and (c), we see that the 3D position of each person can be detected accurately, even when there is overlapping of people who are facing in different directions. The processing times for one frame (361 detection windows) when an Intel Xeon 3-GHz CPU was used are shown in Table 1. Since the proposed method performs the processing in approximately 100 ms, it enables real-time detection at approximately 10 fps.

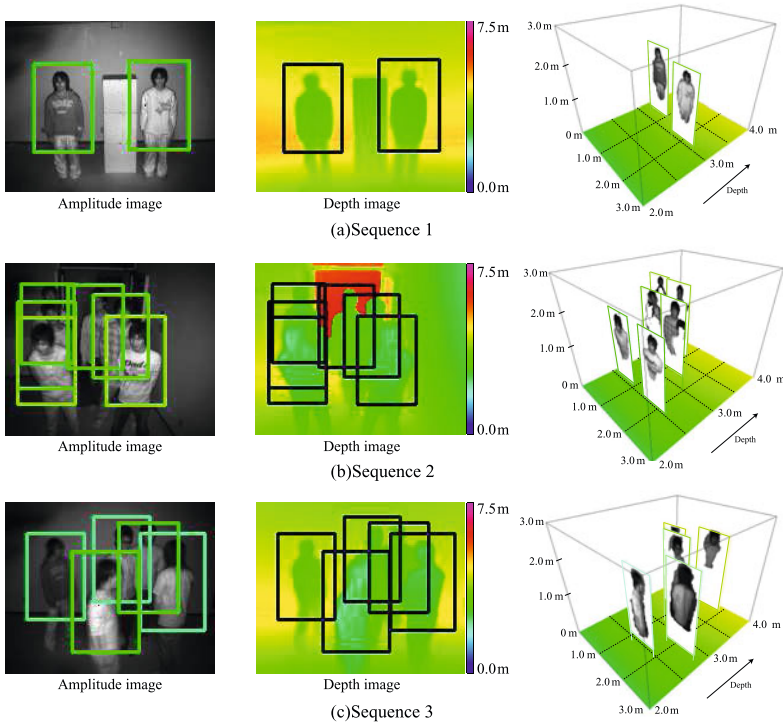


Fig. 14. Example of human detection

5 Conclusions

In this paper, we proposed a real-time human detection method that uses depth information obtained from a TOF camera and can cope with overlapping people and complex scenes. The results of evaluation experiments show that this method enables a 11.5% improvement in detection rate over the use of HOG features, which is a conventional method. In addition, we have confirmed that the proposed method enables highly-accurate classifications even when occlusions occur, by calculating the occlusion rate within the detection window, and performing classifications from consideration of occlusion regions. Since the proposed method requires a total processing time of only approximately 100 ms for the computation and classification of features and the integration of detection windows, it enables real-time detection at approximately 10 fps. In the future, we plan to perform motion analysis using depth information and its variation with time.

References

1. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, vol. 1, pp. 886–893 (2005)

2. Levi, K., Weiss, Y.: Learning object detection from a small number of examples: the importance of good features. In: CVPR, vol. 2, pp. 53–60 (2004)
3. Mitsui, T., Fujiyoshi, H.: Object detection by joint features based on two-stage boosting. In: International Workshop on Visual Surveillance, pp. 1169–1176 (2009)
4. Sabzmeydani, P., Mori, G.: Detecting pedestrians by learning shapelet feature. In: CVPR, pp. 511–518 (2007)
5. Wu, B., Nevatia, R.: Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In: ICCV, vol. 1, pp. 90–97 (2005)
6. Ess, A., Leibe, B., Van Gool, L.J.: Depth and appearance for mobile scene analysis. In: ICCV (2007)
7. Bhattacharyya, A.: On a measure of divergence between two statistical populations defined by probability distributions. Bull. Calcutta Math. Soc. 35, 99–109 (1943)
8. Porikli, F.: Integral histogram: A fast way to extract histograms in cartesian spaces. In: CVPR, pp. 829–836 (2005)
9. Schapire, R.E., Singer, Y.: Improved boosting algorithms using confidence-rated predictions. Machine Learning 37, 297–336 (1999)
10. Mure-Dubois, J., Hugli, H.: Fusion of time of flight camera point clouds. In: ECCV Workshop on M2SFA2 (2008)
11. Comaniciu, D., Meer, P.: Mean shift analysis and applications. In: ICCV, pp. 1197–1203 (1999)

Human Tracking by Multiple Kernel Boosting with Locality Affinity Constraints

Fan Yang¹, Huchuan Lu¹, and Yen-Wei Chen^{1,2}

¹ School of Information and Communication Engineering,
Dalian University of Technology, Dalian, China

² College of Information Science and Engineering,
Ritsumeikan University, Kusatsu, Japan

Abstract. In this paper, we incorporate the concept of Multiple Kernel Learning (MKL) algorithm, which is used in object categorization, into human tracking field. For efficiency, we devise an algorithm called Multiple Kernel Boosting (MKB), instead of directly adopting MKL. MKB aims to find an optimal combination of many single kernel SVMs focusing on different features and kernels by boosting technique. Besides, we apply Locality Affinity Constraints (LAC) to each selected SVM. LAC is computed from the distribution of support vectors of respective SVM, recording the underlying locality of training data. An update scheme to reselect good SVMs, adjust their weights and recalculate LAC is also included. Experiments on standard and our own testing sequences show that our MKB tracking outperforms some other state-of-the-art algorithms in handling various conditions.

1 Introduction

Visual tracking has been popular in the computer vision community for decades. In this paper, we consider tracking as a binary classification, aiming to discriminate the object from the background in successive frames. Collins *et al.* [1] propose a method to adaptively select color features that best separate the object from the background. Grabner *et al.* [2] design an online version of Adaboost classifier for object tracking, which accumulates samples to train a strong classifier and then use the classifier to find the object in videos. To solve drifting problem, SemiBoost tracker [3], also a boosting classifier combined with semi-supervised learning, is proposed. Avidan proposes support vector tracking (SVT) [4] which utilizes an off-line SVM to discriminate the target vehicle from the background, and an ensemble tracking approach [5]. The main concept of “ensemble” is to collect a number of weak classifiers to learn the difference between the object and the background, and then iteratively train new weak classifiers to replace old ones. Tian *et al.* [6] devise an ensemble SVM classifier based tracking algorithm. They use linear SVM to automatically select “key frame” of the target as support vectors. By combining several linear SVM classifiers, history information is

integrated into the tracking framework. More recently, Babenko *et al.* [7] propose a tracking framework utilizing Multiple Instance Learning (MIL) algorithm to augment training and update samples.

Noticing that the SVM-based classifier can effectively solve classification problem in tracking field, we focus on the kernel learning technique used in object classification. The basic idea of kernel used in non-linear SVM is to map training samples from the input space to a higher dimensional feature space, where they are linearly separable, without explicitly defining the mapping function. In particular, we are interested in Multiple Kernel Learning (MKL) [8,9,10], which has shown great advantages in the recent object classification task [11,12]. MKL aims to learn an optimal kernel combination and assign appropriate weight to each kernel in supervised learning settings. Standard MKL displays remarkable ability to solve multi-class classification problems. However, for better classification, many improvements have been proposed. Rakotomamonjy *et al.* [8] propose an improved MKL algorithm, named SimpleMKL, for simplifying the optimization process based on mixed-norm regularization. Localized MKL (LMKL) [13] and Bayesian Localized MKL (BLMKL) [14] are devised to exploit the distribution of training data on each kernel space and give higher weights to appropriate kernel functions if data has underlying localities. Motivated by LMKL, Cao *et al.* [15] propose Heterogeneous Feature Machines (HFM) to learn a non-linear combination of multiple kernels; Yang *et al.* [16] propose group-sensitive multiple kernel learning (GS-MKL) to accommodate the intra-class diversity and the inter-class correlation for object categorization. Boosting method is also incorporated into MKL to implement feature combination [17] and feature selection [18].

Impressed by the remarkable performance of MKL, we propose a Multiple Kernel Boosting (MKB) algorithm with Locality Affinity Constraints (LAC) for human tracking. To describe an object, we use 3 feature descriptors, RGB histogram, Histogram of Gradient (HoG) [19] and SIFT [20]; to map the input space to the kernel space, we use 4 kernels, linear kernel, polynomial kernel, RBF kernel and sigmoid kernel. We consider each single kernel SVM as a “weak classifier”. To find the best combination of these SVMs, we utilize boosting technique instead of a global optimization used in most MKL algorithms. We also introduce locality affinity information of input data, which is computed from the distribution of support vectors of the respective single kernel SVM, into the final decision function. In each new frame, we apply particle sampling to generate a number of candidates. Tracking is then accomplished by finding the best candidate. For update, we retrain the set of single kernel SVMs, reselect some discriminative ones by MKB, and recalculate LAC.

The remainder of the paper is organized as follows: Section 2 and Section 3 introduce our Multiple Kernel Boosting (MKB) algorithm and Locality Affinity Constraints (LAC) respectively. Main tracking framework is in Section 4 and experimental results on various sequences are shown and discussed in Section 5. The last section gives out conclusion.

2 Multiple Kernel Boosting

2.1 Standard MKL

The main difficulty of single SVM is to choose a proper kernel for the given training dataset. However, MKL aims to find an optimal convex combination of multiple kernels and the associated classifier simultaneously. For binary classification, assuming that we have training samples $\{x_i, y_i\}_{i=1}^D$, where x_i is the i^{th} sample and $y_i = \{\pm 1\}$ indicates the label of the sample, our task is to train a multi-kernel based classifier $F(x)$ to classify an unlabeled sample into a class. Let $\{K_m\}_{m=1}^M$ be the kernel matrices computed for different feature modalities. The combination of multiple kernels is defined as

$$K(x, x_i) = \sum_{m=1}^M \beta_m K_m(x, x_i) \quad (1)$$

where kernel weights $\beta_m \geq 0$ and $\sum_{m=1}^M \beta_m = 1$. K_m can be the same kernels with different hyperparameters or different kernels. Also, they can be applied to different feature sets. Then the decision function is defined as

$$F(x) = \sum_{i=1}^D \alpha_i y_i \sum_{m=1}^M \beta_m K_m(x, x_i) + b \quad (2)$$

where $\{\alpha_i\}$ and b are the Lagrange multipliers and the bias in the standard SVM algorithm. We can learn $\{\alpha_i\}$, $\{\beta_m\}$ and b from a joint optimization process. Details can be found in [10].

2.2 Multiple Kernel Boosting

Despite its success in object categorization, MKL cannot be directly applied to tracking due to time-consuming optimization process, large amount of training samples and constant weights. However, Gehler and Nowozin [17] have discussed a boosting version of MKL for feature combination, which inspires us to propose Multiple Kernel Boosting (MKB) for tracking applications. For a sample x , we construct a vector by concatenating its kernel values with all the training samples $\{x_i, y_i\}_{i=1}^D$ to indicate the m^{th} kernel response

$$K_m(x) = [K_m(x, x_1), K_m(x, x_2), \dots, K_m(x, x_D)]^T \quad (3)$$

So we can rewrite Equation 2 as the following form

$$\begin{aligned} F(x) &= \sum_{m=1}^M \beta_m \sum_{i=1}^D \alpha_i y_i K_m(x, x_i) + b \\ &= \sum_{m=1}^M \beta_m (K_m(x)^T \alpha + b) \end{aligned} \quad (4)$$

where $\alpha = (\alpha_1 y_1, \alpha_2 y_2, \dots, \alpha_D y_D)^T$. So we convert standard MKL to a linear combination of the real value output of M separate SVMs $K_m(x)^T \alpha + b$. According to [17], we can separately train M SVMs with different parameters $\{\alpha_m, b_m\}$ at first, and then optimize $\{\beta_m\}$ in the second step. Each individual SVM is not restricted to share the same parameter. By letting $h_m(x) = K_m(x)^T \alpha + b$, we convert the decision function of standard MKL to $F(x) = \sum_{m=1}^M \beta_m h_m(x)$. To determine $\{\beta_m\}$, we can simply use other methods. In this paper, we use boosting method, so we name our algorithm Multiple Kernel Boosting (MKB). In the boosting form, the decision function can be written as

$$F(x) = \sum_{l=1}^L \beta_l h_l(x) \quad (5)$$

where L indicates the iteration time. We regard MKL as choosing multiple “weak” single kernel SVMs into a final strong classifier. MKB avoids complex global optimization, thereby making the concept of MKL applicable to tracking.

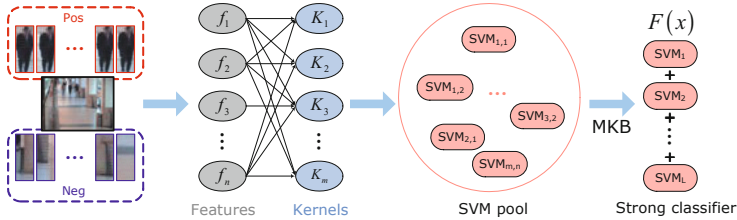


Fig. 1. Illustration of Multiple Kernel Boosting (MKB) process

As Figure 1 shows, we extract $\{f_1, f_2, \dots, f_N\}$ features from a set of positive and negative samples and send them into $\{K_1, K_2, \dots, K_M\}$ kernels. Then we get $M \times N$ combinations; for each combination, we train a single kernel SVM. The classification error of a single kernel SVM is defined as

$$\varepsilon = \frac{\sum_{i=1}^D w(i) \cdot |h(x_i)| \cdot U(y_i h(x_i))}{\sum_{i=1}^D w(i) \cdot |h(x_i)| \cdot U(-y_i h(x_i))} \quad (6)$$

Here $U(x)$ is a function that equals 1 when $x > 0$, otherwise it equals 0. $w(i)$ is training samples’ weight. $h(x_i)$ is the real value classification output of the SVM on the input x_i . We aim to adaptively select multiple features and kernels that are of the most discriminative ability from the pool. So we use boosting technique to iteratively choose an SVM and add it to the final decision function. The complete process of MKB is shown in Algorithm 1.

Algorithm 1. Multiple Kernel Boosting (MKB)

Input: training sets $\{x_i, y_i\}_{i=1}^D$, feature functions $\{f_n\}_{n=1}^N$, kernel functions $\{K_m\}_{m=1}^M$, the decision function $F(x) = 0$

1: for each $n \in N$ and $m \in M$, train a single kernel SVM $h_{m,n}(x)$ on feature f_n and kernel K_m on the entire training set $\{x_i, y_i\}_{i=1}^D$ to form a pool of candidate single kernel SVMs, denoted as h

2: initialize samples' weights $w_1(i) = 1/D$

3: for $l = 1$ to L do

1) For each $h_{m,n}(x)$, compute classification error $\varepsilon_{m,n}$ using Equation 6

2) Select $h_l(x) = \arg \min_{h_{m,n} \in h} \varepsilon_{m,n}$

3) Compute weight $\beta_l = \frac{1}{2} \log \frac{1-\varepsilon_l}{\varepsilon_l}$ for $h_l(x)$

4) If $\beta_l < 0$, break; otherwise add $h_l(x)$ to $F(x) \leftarrow F(x) + \beta_l h_l(x)$

5) $w_{l+1}(i) = \frac{w_l(i)}{Z_l} e^{-\beta_l y_i h_l(x_i)}$

4: end for

Output: final strong classifier $F(x) = \sum_{l=1}^L \beta_l h_l(x)$

3 Locality Affinity Constraints

Although MKB produces promising tracking results, we find that it is not stable enough in some cases. So we try to improve the original MKB. Motivated by LMKL [13] and GS-MKL [16], we incorporate the distribution of training data into $F(x)$ to enhance the robustness of MKB. Rewriting Equation 2, we obtain

$$F(x) = \sum_{i=1}^D \alpha_i y_i \sum_{m=1}^M \beta_m(x) K_m(x, x_i) + b \quad (7)$$

where $\beta_m(x)$ is a function of input x , rather than a constant β_m in the standard MKL. It can be learned from an iteration algorithm [13]. However, we find that the optimization process is intolerantly time-consuming [16]. Moreover, iteration cannot guarantee convergence to global optimum and unsuitable initial parameters may also degrade the performance. Considering the problem of limited training samples in tracking, we devise a simple but effective method to exploit the underlying distribution of training data.

We assume that an SVM trained in MKB has recorded the property of training data with respect to the feature and kernel. Since support vectors of each SVM reserve most information, we utilize those support vectors for computing the locality of data. Letting $\beta_l = \beta_l^* A_l(x)$ and rewriting Equation 5, we get

$$F(x) = \sum_{l=1}^L \beta_l^* A_l(x) h_l(x) \quad (8)$$

where β_l^* is the same as β_l in Equation 5, which is calculated by MKB. $A_l(x)$ is a function of input x , indicating the similarity of x with the trained SVM, which is called Locality Affinity Constraint (LAC) in our algorithm. Locality

affinity means that if the input sample complies with the distribution of support vectors in a specific SVM, we think that the importance of the corresponding SVM is high, thus assigning it larger weight. We construct a probability model to describe the locality affinity, which is defined as

$$A_l(x) = 1 - \exp(-|\sigma_l(x)|) \quad (9)$$

where $\sigma_l(x) = \log \left[\frac{p_l(y=1|x)}{p_l(y=-1|x)} \right]$. For each trained SVM $h_l(x)$, we compute the mean μ_l^+ and μ_l^- of positive and negative support vectors respectively. Then $p_l(y=1|x)$ and $p_l(y=-1|x)$ are computed as follows

$$p_l(y^*|x) = \exp(-|x - \mu_l^*|) \quad (10)$$

where $y^* = 1$ or $y^* = -1$ when μ_l^* is μ_l^+ or μ_l^- . Here, $A_l(x) \in (0, 1)$, which can be seen as the probability of sample x belonging to the support vectors. If x is similar with training data on a specific combination of feature and kernel, the importance of the corresponding SVM is high, and vice versa. Therefore, we formulate the distribution of training samples and impose such constraints on testing samples, thereby improving the discriminative ability of the decision function.

4 Main Tracking Framework

In this section, we will introduce how tracking proceeds based on the aforementioned algorithms. In 1st frame, we draw a bounding box x^1 enclosing the object we want to track, where $x^1 = (c_x^1, c_y^1, s^1, \theta^1)$ records the center, size and rotation angle of the object. The superscript indicates the current frame number. To augment the number of training samples, we crop out a set of images $X^+ = \{x_i | 0 \leq l(x_i) - l(x^1) < r_\alpha\}_{i=1}^{D^+}$ to collect positive samples. Here r_α is a small constant and $l(x)$ indicates the center of x . Similarly, we crop out a set of negative samples $X^- = \{x_i | r_\beta \leq l(x_i) - l(x^1) < r_\gamma\}_{i=1}^{D^-}$. We set $r_\beta > r_\alpha$ to allow less than 1/4 overlap between positive and negative samples. Note that we only use 1st frame to collect $(D^+ + D^-)$ training samples. Extracting features on these samples, performing MKB and adding locality affinity functions, we obtain a multi-kernel based decision function.

To improve efficiency, we adopt particle sampling technique in the following frames. The predicting distribution of x^t given all available observations $z^{1:t-1} = \{z^1, z^2, \dots, z^{t-1}\}$, denoted by $p(x^t|z^{1:t-1})$, is recursively computed as

$$p(x^t|z^{1:t-1}) = \int p(x^t|x^{t-1})p(x^{t-1}|z^{1:t-1})dx^{t-1} \quad (11)$$

When the observation z^t is obtained at time t , the state vector is updated as

$$p(x^t|z^{1:t}) = \frac{p(z^t|x^t)p(x^t|z^{1:t-1})}{p(z^t|z^{1:t-1})} \quad (12)$$

Algorithm 2. MKB Tracking with Locality Affinity Constraints

Input: training sets $\{x_i, y_i\}_{i=1}^D$, feature functions $\{f_n\}_{n=1}^N$, kernel functions $\{K_m\}_{m=1}^M$, the decision function $F(x) = 0$, empty sample queue Q

Output: tracking results in each frame $\{x^1, x^2, \dots, x^t\}$

For the first frame I_t ($t = 1$)

- 1: given the bounding box $x^1 = (c_x^1, c_y^1, s^1, \theta^1)$, extract D^+ positive samples and D^- negative samples
- 2: extract features $\{f_n(x_i)\}_{i=1}^{D^+ + D^-}$ and train individual single kernel SVMs $h_{m,n}(x)$
- 3: compute the locality affinity function $A_{m,n}(x)$ according to the distribution of support vectors for each trained SVM $h_{m,n}(x)$
4. apply **Algorithm 1** to obtain the strong classifier $F(x) = \sum_{l=1}^L \beta_l^* A_l(x) h_l(x)$

For each new frame I_t ($t > 1$)

- 1: sample D particles $\{x_i^t\}_{i=1}^D$ around the tracked object x^{t-1} according to distribution $p(x^t | x^{t-1})$. The weight of each particles $\{w_i^t = 1\}_{i=1}^D$
 - 2: use $F(x)$ to compute classification results of $\{x_i^t\}_{i=1}^D$, then $\{w_i^t = \exp(F(x_i^t)) / Z^t\}_{i=1}^D$, where Z^t is a normalized value
 - 3: the tracked object is find by $x^t = \sum_{i=1}^D w_i^t x_i^t$
 - 4: regard x^t as positive sample and collect 4 negative samples around x^t , push them into the sample queue Q
 - 5: if the length of sample queue $Length(Q) = 5T_u$, do
 - 1) select SVM $h_{m,n}(x)$ from weak SVM pool h, extract feature $S_Q = f_n(x), x \in Q$. Form new training sample groups $S'_{m,n} = S_{m,n} \cup S_Q$, where $S_{m,n}$ are support vectors of $h_{m,n}(x)$. Train $h_{m,n}(x)$ again using $S'_{m,n}$
 - 2) remove $h_{m,n}(x)$ from the pool h
 - 3) repeat 1) and 2) until the pool is empty
 - 4) update $\mu_{m,n}^+$ and $\mu_{m,n}^-$ of new trained support vectors to obtain new $A_{m,n}(x)$
 - 5) perform **Algorithm 1** again to reselect appropriate $h_l(x)$ to form a new

$$F(x) = \sum_{l=1}^L \beta_l^* A_l(x) h_l(x)$$
 - 6) clean up the sample queue Q
 - 6: otherwise output x^t and proceed to the next frame
-

where $p(z^t | x^t)$ is the observation likelihood. The posterior probability $p(x^t | z^{1:t})$ is approximated by D particles $\{x_i^t\}_{i=1}^D$ with importance weight w_i^t , which are drawn from a reference distribution $q(x^t | x^{1:t-1}, z^{1:t})$. We let $q(x^t | x^{1:t-1}, z^{1:t}) = p(x^t | x^{t-1})$ then the weights $w_i^t = w_i^{t-1} p(z^t | x_i^t)$. We think that $p(x^t | x^{t-1})$ complies with a Gaussian distribution and affine parameters in x^t are independent. So in frame I_t , we have D candidates with different affine parameters around the tracked object x^{t-1} in frame I_{t-1} . Then we apply $p(z^t | x_i^t) = e^{F(x_i^t)}$ to compute $p(z^t | x_i^t)$, which is particle's weight. Subsequently, we normalize $\{w_i^t\}_{i=1}^D$ and compute the weighted sum of particles to find the object, denoted as $x^t = \sum_{i=1}^D w_i^t x_i^t$.

Moreover, to capture the variance of the object, we also incorporate an update scheme into the tracking framework. In each frame, we consider the tracked

object x^t as positive sample, and extract four negative samples from four directions (up, down, left, right) without overlap with the positive one. We accumulate these samples for T_u frames, then retrain individual SVMs using new samples and corresponding support vectors. Subsequently, we perform MKB again to obtain a new $F(x)$. $A_l(x)$ is also recalculated from new support vectors. The complete process is shown in Algorithm 2.

5 Experiments

5.1 Experimental Settings

We implement our tracking algorithm by Matlab. In 1st frame, both positive and negative samples are 20. The pool of weak SVMs contains 12 single kernel SVMs, each of which focuses on a specific combination of 3 features (64-dim RGB histogram, 128-dim HoG and 128-dim SIFT descriptor) and 4 kernels (linear kernel, polynomial kernel, RBF kernel and sigmoid kernel). The iteration time of MKB is 10, while the number of selected SVMs varies according to different sequences. In each new frame, we sample 200 particles according to a pre-defined distribution and send them to $F(x)$ to get 200 real values. The update rate also varies according to the property of different sequences. Note that all parameters are fixed except the distribution for sampling affine parameters of sequences.

We also run other three tracking systems: Online Adaboost tracking (OAB) [2], Multiple Instance Learning tracking (MIL) [7] and color-based particle filter tracking (PF) [21]. Similar with our MKB tracking, both OAB and MIL tracking rely on a boosting technique and use new samples to change weak classifiers and corresponding weights. Also, our approach includes the particle sampling that generates a number of candidates used to approximate the current state of the object. We are going to show that the good performance of our MKB tracking is not necessarily attributed to particle sampling, so our approach outperforms PF in most sequences. In our experiments, the number of selectors in OAB remains 100; PF uses 512-dim RGB feature ($8 \times 8 \times 8$ bins) and its sampling parameters are constant on all sequences.

5.2 Results

We compare our method with OAB, MIL and PF tracking. To better display the advantages of the proposed method, we will analyze the tracking results under various situations.

Occlusion. Figure 2 shows a comparison under occlusion. The testing sequence is from CAVIAR database. Our MKB tracking continuously keeps track of the person even when he is occluded by another person in similar color. While all other methods drift away from the object when occlusion occurs (OAB and MIL) or when the object’s size changes (PF). We also find that HoG plays the most important part when occlusion occurs. Because the color of the two persons is almost the same, color feature is unreliable.



Fig. 2. Comparison of tracking results when there is occlusion. Top row shows results of our approach. Results of other approaches are shown in the bottom row.



Fig. 3. Comparison of tracking results when there is scale change. Top row shows results of our approach. Results of other approaches are shown in the bottom row.

Scale change. To test the ability to handle the object’s scale change, we also compare the four algorithms on another sequence also from CAVIAR database, as shown in Figure 3. In the sequence, a person walks away from the camera, so his size becomes smaller than that in the first few frames. There is also simple occlusion by other people. Our approach can locate the person’s position accurately and the tracking result is quite stable. In contrast, lacking a scheme of adaptively adjusting the size of the tracking window, both OAB and MIL lose the object when large scale change occurs. PF is even confused by the other three persons close to the real object, even though they do not occlude the object.

Complex background. We also run the four algorithms on our own testing sequences. Figure 4 shows tracking results on a sequence, in which a figure skater exhibits a set of actions in a skating rink. As the figure shows, the background is complex, including various colors. Sometimes the dark background is even the same as the skater’s black clothes. MIL, OAB and PF cannot find the precise position of the skater, especially when he changes his poses under the dark background; while PF even loses the skater when he changes his skating direction. In contrast, our approach can locate the skater, resulting in more accurate results. Therefore, our MKB tracking has the ability to deal with complex background.

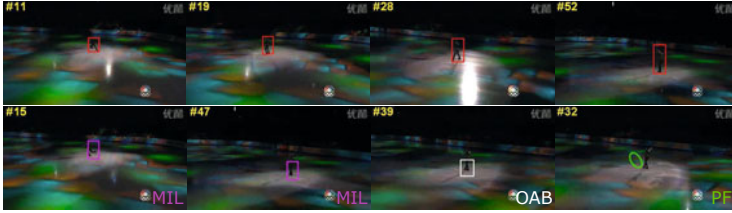


Fig. 4. Comparison of tracking results when background is complex. Top row shows results of our approach. Results of other approaches are shown in the bottom row.



Fig. 5. Comparison of tracking results when there is fast motion. Top row shows results of our approach. Results of other approaches are shown in the bottom row.

Fast motion. The last experiment we will report is to show the ability to handle fast motion of the object. We aim to keep track of the famous sprinter Bolt in the sequence that includes part of a 100m dash competition. The main difficulty is that the runner moves fast. From Figure 5, we can see that both MIL and OAB drifts away just in the first few frames when Bolt starts to accelerate; while our MKB tracking and PF find Bolt accurately until about 120th frame. However, we also find that PF is much sensitive to the sampling parameters: slight change of initial parameters may affect the performance severely. Compared with it, our method is more robust. The change of sampling parameters within an appropriate range does not decrease the accuracy.

5.3 Discussions

In this section, we will briefly discuss some properties of our proposed method. Table 1 shows quantitative comparisons on 7 testing sequences. Numbers indicate the average error of center of the object per frame on testing sequences.

MKB tracking vs. single kernel SVM. First, we compare the proposed algorithm with single kernel SVM using only one feature. We observe that in most sequences, HoG+linear kernel and SIFT+linear kernel perform well. So we compare the tracking results of the two single kernel methods. In Table 1, S1 and S2 indicate HoG+linear kernel and SIFT+linear kernel respectively. From Table 1, we can see that only using one combination of feature and kernel cannot achieve good results. Both the average position errors of the two methods are

Table 1. The average position error per frame

	OAB	MIL	PF	MKB	S1	S2	LAC-
<i>ShopAssistant2cor</i>	67.6	68.2	13.1	<u>4.5</u>	4.8	19.8	2.8
<i>ThreePastShop2cor</i>	15.2	17.7	35.8	3.5	129.8	19.8	<u>4.2</u>
<i>MeetWalkSplit</i>	<u>9.0</u>	21.5	<u>9.0</u>	8.0	12.0	89.4	11.9
<i>skate</i>	20.7	13.8	25.1	12.4	26.3	23.4	<u>13.3</u>
<i>dash</i>	129.4	206.9	<u>11.1</u>	4.1	18.8	12.4	16.1
<i>Browse1</i>	13.6	8.4	36.0	<u>7.5</u>	152.5	108.6	5.4
<i>OneLeaveShopR1cor</i>	<u>7.3</u>	10.7	8.9	4.6	31.8	46.2	51.6

much larger than that of our MKB tracking, although in some cases HoG+linear kernel can produce more accurate results than other state-of-the-art approaches.

MKB tracking vs. other approaches. Besides qualitative comparisons in the previous section, we also give out quantitative comparisons of our proposed tracking and other algorithms. From the table, we can see that our MKB tracking is much more robust. The adaptive selection of kernels and features shows its advantage, compared with other approaches.

Impact of LAC. To test the effectiveness of LAC, we also run our tracking system without such constraints (see ‘‘LAC-’’ column in Table 1). LAC shows predominant advantage in most cases, leading to lower average position error, although in only two sequences it does not outperform MKB tracking without LAC. Therefore, by incorporating LAC, we boost the performance of original MKB tracking. Moreover, we can see that even we use standard MKB for tracking, the results are not inferior to other existing approaches.

6 Conclusion

In this paper, we incorporate the concept of Multiple Kernel Learning (MKL) algorithm, which is used in object categorization, into human tracking field. We devise an algorithm called Multiple Kernel Boosting (MKB), instead of directly adopting MKL. In MKB, we treat individual single kernel SVMs as weak classifiers and utilize boosting technique to adaptively select a number of good SVMs into a final decision function focusing on different features and kernels. Compared with standard MKL, MKB is much efficient. To strengthen the discriminative ability of the strong classifier formed in MKB, we also apply Locality Affinity Constraints (LAC) to each selected SVM. LAC is computed from the distribution of support vectors of respective SVM, recording the underlying locality of training data. An update scheme to reselect good SVMs, adjust their weights and recalculate LAC is also included in our tracking framework. Experiments on some standard and our own testing sequences show that our MKB tracking outperforms some of its rivals in handling simple occlusion, scale change and complex background.

Acknowledgement. The work was supported by the Fundamental Research Funds for the Central Universities, No. DUT10JS05, and the National Natural Science Foundation of China (NSFC), No.61071209.

References

1. Collins, R., Liu, Y., Leordeanu, M.: Online selection of discriminative tracking features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1631–1643 (2005)
2. Grabner, H., Bischof, H.: On-line boosting and vision. In: *CVPR*, vol. 1, pp. 260–267 (2006)
3. Grabner, H., Leistner, C., Bischof, H.: Semi-supervised on-line boosting for robust tracking. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part I. LNCS*, vol. 5302, pp. 234–247. Springer, Heidelberg (2008)
4. Avidan, S.: Support vector tracking. *PAMI* 26(8), 1064–1072 (2004)
5. Avidan, S.: Ensemble tracking. In: *CVPR*, vol. 2, pp. 494–501 (2005)
6. Tian, M., Zhang, W., Liu, F.: On-line ensemble svm for robust object tracking. In: Yagi, Y., Kang, S.B., Kweon, I.S., Zha, H. (eds.) *ACCV 2007, Part I. LNCS*, vol. 4843, pp. 355–364. Springer, Heidelberg (2007)
7. Babenko, B., Yang, M.H., Belongie, S.: Visual tracking with online multiple instance learning. In: *CVPR*, pp. 983–990 (2009)
8. Rakotomamonjy, A., Bach, F., Canu, S., Grandvalet, Y.: SimpleMKL. *Journal of Machine Learning Research* 9, 2491–2521 (2008)
9. Sonnenburg, S., Ratsch, G., Schafer, C., Scholkopf, B.: Large scale multiple kernel learning. *Journal of Machine Learning Research* 7, 1531–1565 (2006)
10. Bach, F., Lanckriet, G., Jordan, M.: Multiple kernel learning, conic duality, and the SMO algorithm. In: *ICML* (2004)
11. Varma, M., Ray, D.: Learning the discriminative power-invariance trade-off. In: *ICCV* (2007)
12. Kumar, A., Sminchisescu, C.: Support kernel machines for object recognition. In: *ICCV* (2007)
13. Gonen, M., Elpaz, E.: Localized multiple kernel learning. In: *ICML*, pp. 352–359 (2008)
14. Christoudias, M., Urtasun, R., Darrell, T.: Bayesian localized multiple kernel learning. Technical Report UCB/EECS-2009-96, EECS Department, University of California, Berkeley (2009)
15. Cao, L., Luo, J., Liang, F., Huang, T.: Heterogeneous Feature Machines for Visual Recognition. In: *ICCV*, pp. 1095–1102 (2009)
16. Yang, J., Li, Y., Tian, Y., Duan, L., Gao, W.: Group-Sensitive Multiple Kernel Learning for Object Categorization. In: *ICCV* (2009)
17. Gehler, P., Nowozin, S.: On feature combination for multiclass object classification. In: *ICCV 2009*, pp. 221–228 (2009)
18. Siddiquie, B., Vitaladevuni, S., Davis, L.: Combining multiple kernels for efficient image classification. In: *WACV 2009*, pp. 1–8 (2009)
19. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: *CVPR*, vol. 1, pp. 886–893 (2005)
20. Lowe, D.: Object recognition from local scale-invariant features. In: *ICCV*, pp. 1150–1157 (1999)
21. Nummiaro, K., Koller-Meier, E., Van Gool, L.: Object tracking with an adaptive color-based particle filter. In: Van Gool, L. (ed.) *DAGM 2002. LNCS*, vol. 2449, pp. 353–360. Springer, Heidelberg (2002)

A Temporal Latent Topic Model for Facial Expression Recognition

Lifeng Shang and Kwok-Ping Chan

The University of Hong Kong, Pokfulam, Hong Kong
{lfs Shang, kpchan}@cs.hku.hk

Abstract. In this paper we extend the latent Dirichlet allocation (LDA) topic model to model facial expression dynamics. Our topic model integrates the temporal information of image sequences through redefining topic generation probability without involving new latent variables or increasing inference difficulties. A collapsed Gibbs sampler is derived for batch learning with labeled training dataset and an efficient learning method for testing data is also discussed. We describe the resulting temporal latent topic model (TLTM) in detail and show how it can be applied to facial expression recognition. Experiments on CMU expression database illustrate that the proposed TLTM is very efficient in facial expression recognition.

1 Introduction

Facial expression recognition has become an active research topic in recent years due to its potential applications in human computer interfaces, data-driven animation, etc. Most facial expression recognition methods attempt to recognize six prototypic expressions (namely joy, surprise, anger, disgust, sadness and fear) proposed by Ekman [6]. Over the past decade, many techniques (e.g. Neural networks [22]) have been applied to still facial images recognition. Psychological studies show that facial image sequences often produce more accurate and robust recognition compared to mug shots [1]. Therefore, recent attention has been moving to model the facial expression dynamics through integrating temporal information [12] [18] [19].

The approaches to modeling temporal behaviors of facial expressions are generally classified as designing dynamic features (e.g. Dynamic Texture [27]) or constructing sequential data modeling tools (e.g. Dynamic Graphical Model [26]). Yang et al. [24] designed a dynamic Haar-like feature to represent facial image sequences. Zhao et al. [27] extended the well-known local binary feature (LBP) to the temporal domain and applied it to facial expression recognition. Yeasin et al. [25] captured the dynamics of facial image sequences by Hidden Markov Models (HMMs). To better model the relative change of emotional magnitude, Zhang et al. [26] presented a probabilistic framework by integrating the Dynamic Bayesian networks (DBNs) with the facial action units (AUs) [6]. Their methods can reflect the evolution of a spontaneous expression. DBNs are natural for modeling facial expression variations, and can be easily extended by combining them

with other models (e.g. Neural Networks) or incorporating semantic relationships between AUs. Nevertheless, modeling the temporal order of facial expression explicitly is risky, because noise in the facial features can easily propagate through the model. Moreover, these models often suffer from too many latent variables or too complex model structures, which makes learning and inference difficult.

Recently, in the statistical text community latent topic models (e.g. LDA [2]) have achieved significant success in semantic clustering. Besides modeling text generation, LDA has also been widely used to solve computer vision problems, e.g. object discovery [23] and scene categorization [15]. However, directly applying a language model to computer vision problems has some difficulties, since in LDA the “bag-of-words” representation relies on the assumption that the order of words or documents can both be ignored. As pointed out by Wang et al. [23], the spatial and temporal structure of documents or words are meaningless in a language model, but important for many computer vision problems. Therefore, studies on extending the LDA to model the spatiotemporal structures of words, topics, documents or corpora have gained more and more attention. Wang et al. [23] proposed a spatial LDA to include the spatiotemporal structure among visual words. Hanna [8] considered word order information by incorporating n -gram statistics. Hospedales et al. [9] combined HMM with LDA to model behavior dynamics. In this paper, we propose a new latent topic model (TLTM) which considers the temporal structure of facial image sequences. In TLTM, facial expression dynamics is included by redefining topic generation probability to ensure that successive images are most likely to have the similar topic distributions. Compared to existing extensions, our TLTM does not use new latent variables nor increase inference difficulties, which makes it as efficient as LDA. Experiments on CMU facial expression dataset [11] show that our generative TLTM model outperforms the generally used HMM models and achieves comparable performance as some discriminant models.

The rest of this paper is organized as follows. In Section 2, we describe the feature extraction method. In Section 3, we introduce the proposed TLTM and apply it to facial expression recognition. In Section 4, the performance of proposed method is evaluated by the CMU dataset. Section 5 summarizes this paper.

2 Feature Extraction and Indication

In facial expression recognition, there are two types of facial features: permanent and transient features. The permanent facial features are the shapes and locations of facial components (e.g. eyebrows, eye lids, nose, lips and chin). The transient features are the wrinkles and bulges appeared with expressions. In this work, we do not consider transient features and use the movement of facial features away from neutral positions to measure facial expression variation.

We applied the well-known Active Appearance Model (AAM) [5] on facial image sequences to track the movement of facial features. Figure 1(a) shows the shape model consisting of 58 facial points which is identical with the one given in [4]. Figure 2 displays the facial feature localization results of one subject’s six

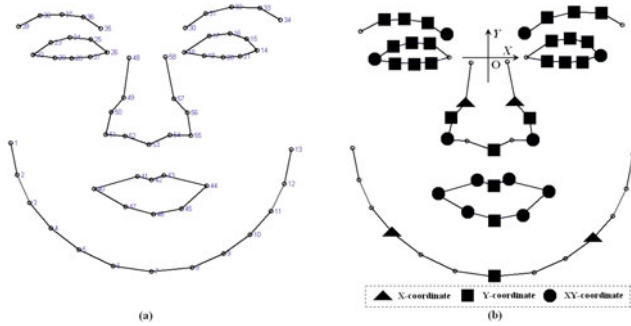


Fig. 1. (a) The facial landmarks(58 facial points) and (b) selected feature points

basic expressions. In [4], the (x, y) coordinates of the 58 localized facial points forming a 116-dimensional vector are used to represent an image. Based on the analysis of facial action coding system (FACS) [6], we found that the movements of some facial points (e.g. facial points 1 and 13) are not essential to measuring facial deformation, so a subset is selected from the 58 facial points as feature points which are depicted in Fig. 1(b), in which the solid triangles and rectangles represent that only the X or Y-coordinates are used as feature and the solid circles represent that both the X and Y-coordinates are used. The midpoint of the inner corners of the two eyes (facial points 18 and 26) is defined as the origin. A facial image is thus represented by a 52-dimensional feature vector.



Fig. 2. The tracking results of one subject's six basic expressions

To further reduce the inter-personal variations with regard to the amplitudes of facial actions, feature points are quantized into a fixed number of words according to movements away from neutral positions. The movement in the X-axis direction is quantized into a word of the vocabulary $\text{VocabularyX} = \{\text{Left}_i, \text{Right}_i, \text{MotionlessX}_i | i = 1, 2, \dots, 58\}$, where the word Left_i (Right_i) represents that the i -th facial point moves at least two pixels left (right) to its neutral position, otherwise it will be quantized to the word MotionlessX_i . Similarly, the vocabulary describing the movement types in the Y-axis direction is defined as $\text{VocabularyY} = \{\text{Up}_i, \text{Down}_i, \text{MotionlessY}_i | i = 1, 2, \dots, 58\}$. With the two vocabularies at hand, for a given facial image d_i , its 52-dimensional feature vector is changed to a bag-of-words representation $\{w_{i,1}, \dots, w_{i,52}\}$. Our image collection (corpus) is constructed by concatenating these bag-of-words representations one after the other.

3 TLTM for Facial Expression Recognition

Facial expressions can be described by the FACS, in which each expression is characterized by the co-occurrence of atomic facial AUs which are represented by some low-level features. LDA is a hierarchical generative topic model, which is very suitable for discovering the co-occurrence of low-level visual words (or higher-level topics). We can find there is a good correspondence between the FACS and LDA model. When LDA is applied to modeling facial expression variations the low-level visual words (i.e. the movements of feature points away from neutral positions) are clustered into higher level topics which correspond to atomic facial action units. In this section we will first briefly review LDA and establish notations, then particularly study how to extend LDA to model facial expression dynamics.

3.1 Latent Dirichlet Allocation

LDA is a generative model for topic discovery which has attracted a lot of interest from the field of machine learning, language processing and computer vision community. Figure 3 shows the graphical model of LDA. In this model, documents are represented as random mixtures over latent topics, which are characterized by discrete distributions over words.

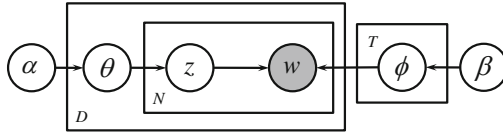


Fig. 3. Plate notation for LDA

Each individual word token w_n in a corpus $\mathbf{w} = \{w_1, w_2, \dots, w_N\}$ is assumed to have been generated by a latent topic z_n , which is drawn from a document-specific distribution over T topics. The probability of generating a word w from a topic t is defined by $\phi_{w|t} = P(w_n = w | z_n = t)$. These probabilities are recorded by a $T \times W$ matrix Φ , where W is the size of vocabulary and T is the number of topics. Similarly, the topic generation is characterized by another conditional probability $\theta_{t|d} = P(z_n = t | d_n = d)$. These probabilities are recorded by a $D \times T$ matrix Θ , where D is the number of documents in the corpus. Thus the joint probability of the corpus \mathbf{w} and a set of corresponding latent topics $\mathbf{z} = \{z_1, \dots, z_N\}$ is

$$P(\mathbf{w}, \mathbf{z} | \Phi, \Theta) = \prod_{n=1}^N \phi_{w_n | z_n} \theta_{z_n | d_n} \quad (1)$$

where w_n is the n -th word of the corpus \mathbf{w} , z_n is the topic assignment for the n -th word and d_n is the document number of the n -th word.

To make the model fully Bayesian, symmetric Dirichlet priors with hyperparameters α and β are placed over Θ and Φ

$$P(\Theta|\alpha) = \prod_d \text{Dirichlet}(\theta_d|\alpha) \text{ and } P(\Phi|\beta) = \prod_t \text{Dirichlet}(\phi_t|\beta) \quad (2)$$

where θ_d is the d -th row of the matrix Θ , ϕ_t is the t -th row of the matrix Φ . Combining the two priors with equation (1) and integrating over Θ and Φ gives the joint probability of corpus and latent topics given hyperparameters: $P(\mathbf{w}, \mathbf{z}|\alpha, \beta)$. Consequently the posterior probability for latent topics \mathbf{z} is calculated

$$P(\mathbf{z}|\mathbf{w}, \alpha, \beta) = \frac{P(\mathbf{w}, \mathbf{z}|\alpha, \beta)}{\sum_{\mathbf{z}} P(\mathbf{w}, \mathbf{z}|\alpha, \beta)}. \quad (3)$$

Unfortunately, exact inference is intractable for LDA, since computing the equation (3) involves evaluating a probability distribution on a large discrete state space. However, there have been three approximating methods to learn LDA, EM with variation inference [2], EM with expectation propagation [16], and Gibbs sampling [7]. In this work, we adopted Gibbs sampling, since this method is better tolerant to local optima and its performance is comparable with the other two methods.

To sample from the posterior distribution (3) using the Gibbs sampling method, we need the full conditional distribution

$$P(z_n = t | \mathbf{z}_{-n}, \mathbf{w}, \alpha, \beta) \propto \frac{N_{-n,t}^{(w_n)} + \beta}{N_{-n,t}^{(\cdot)} + W\beta} \frac{N_{-n,t}^{(d_n)} + \alpha}{N_{-n}^{(d_n)} + T\alpha} \quad (4)$$

where \mathbf{z}_{-n} denotes all the z_j with $j \neq n$, $N_{-n,t}^{(w_n)}$ is the number of times the word w_n assigned to topic t and $N_{-n,t}^{(\cdot)}$ is the number of words assigned to topic t . $N_{-n,t}^{(d_n)}$ is the number of times topic t occurring in document d_n and $N_{-n}^{(d_n)}$ is the number of words in document d_n . All the four numbers do not include the current assignment of z_n . With a set of samples the parameters Θ and Φ can be estimated from \mathbf{w} and \mathbf{z} by

$$\hat{\theta}_{t|d} = \frac{N_t^{(d)} + \alpha}{N^{(d)} + T\alpha}, \text{ and } \hat{\phi}_{w|t} = \frac{N_t^{(w)} + \beta}{N_t^{(\cdot)} + W\beta}. \quad (5)$$

In the context of facial expression recognition, low-level visual words are clustered into higher level topics by LDA which correspond to atomic facial action units. In the next section, our TLTM model will be built based on LDA by including temporal information of image sequences.

3.2 Temporal Latent Topic Model

Before using LDA to model facial expressions dynamics, we need to first define the meaning of ‘‘document’’ for facial expression recognition. If we treat each facial image sequence as a document, the document order information will be

changed to word order information. However, in the standard LDA words are exchangeable, so document structure information will be ignored. To include word order information, Hanna [8] incorporated n -gram statistics. If we define each image as a document, LDA still misses document order information, since LDA is developed for unstructured documents. In [9], Hospedales et al. introduced a Markov chain to model the temporal structure of image sequences. In TLTM, we adopt the latter way that the bag-of-words representation of one facial image is defined as a document. In order to include the temporal information of facial image sequences, we modify the topic generation probability $\theta_{t|d}$ to be $\theta_{t|d, \text{pre}(d)}$, where $\text{pre}(d)$ is the index of the previous image of the d -th image. Since our image collection is constructed by stacking image sequences one after the other and preserving the inner sequence structure, the value of $\text{pre}(d)$ will be $(d - 1)$ or null if image d is the first slice of a sequence. In the case of null, the topic generation probability will be reduced to $\theta_{t|d}$.

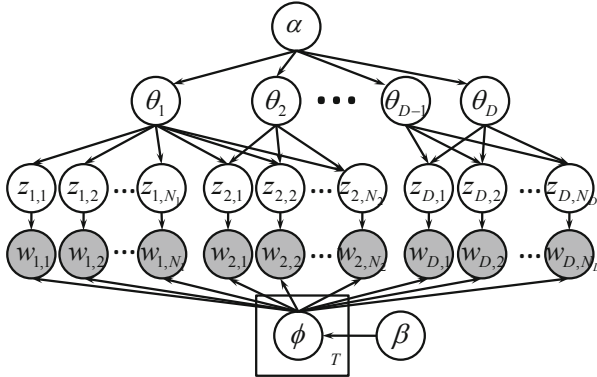


Fig. 4. The graphical model of TLTM

Figure 4 shows the graphical model of TLTM, in which the repeated topic and word generation within the corpus is explicitly drawn. In this figure, N_d is the number of words in the d -th document (image) and has the value of 52 in this work. $w_{i,j}$ is the j -th word of the i -th document and its topic assignment is $z_{i,j}$. The index of the word $w_{i,j}$ in the whole corpus is calculated as $n = ((i - 1) \times 52 + j)$. Compared to the standard LDA as shown in Fig. 3, it can be observed that the topic generation probability $\theta_{t|d}$ does not only depend on θ_d but also depends on $\theta_{(d-1)}$. Use the topic assignment of the word $w_{2,1}$ as an example, the generation probability for $z_{2,1}$ depends both on θ_1 and θ_2 . The generation probability of $z_{2,1}$ is changed from $P(z_{2,1}|\theta_2)$ to $P(z_{2,1}|\theta_1, \theta_2)$. The joint probability $P(\mathbf{w}, \mathbf{z}|\Phi, \Theta)$ becomes to

$$P(\mathbf{w}, \mathbf{z}|\Phi, \Theta) = \prod_{n=1}^N \phi_{w_n|z_n} \theta_{z_n|d_n, (d_n-1)}. \tag{6}$$

According to the Bayes'rule, the distribution $\theta_{t|d,(d-1)}$ can be calculated from the distributions $\theta_{t|d}$ and $\theta_{t|(d-1)}$ as follows

$$\begin{aligned}
 P(t|\theta_d, \theta_{(d-1)}) &= \frac{P(t)P(\theta_d, \theta_{(d-1)}|t)}{P(\theta_d, \theta_{(d-1)})} = \frac{P(t)P(\theta_d|t)P(\theta_{(d-1)}|t)}{P(\theta_d, \theta_{(d-1)})} \\
 &= \frac{P(\theta_d)P(\theta_{(d-1)})}{P(\theta_d, \theta_{(d-1)})} \frac{P(t|\theta_d)P(t|\theta_{(d-1)})}{P(t)} \\
 &\propto \frac{P(t|\theta_d)P(t|\theta_{(d-1)})}{P(t)}, \tag{7}
 \end{aligned}$$

where $P(t)$ is the prior probability of topic t and the prior knowledge here is defined as the set of words, which have the same sequence number as w_n does, and their corresponding topic assignments. So the prior probability $P(t)$ can be regarded as a sequence level topic generation probability with respect to document level topic generation probability (i.e. $\theta_{t|d}$), and it is characterized by a conditional probability $\psi_{t|s} = P(z_n = t | s_n = s)$. These probabilities are recorded by a $S \times T$ matrix Ψ , where S is the number of sequences in the image collection.

As in LDA, we place symmetric Dirichlet priors with hyper parameters α , β and γ over Θ , Φ and Ψ , respectively. $P(\Theta|\alpha)$ and $P(\Phi|\beta)$ are given as in equation (2). $P(\Psi|\gamma)$ is given as follows

$$P(\Psi|\gamma) = \prod_s \text{Dirichlet}(\psi_s|\gamma) \tag{8}$$

where ψ_s is the s -th row of the matrix Ψ . Combining the three priors with equation (6) and integrating over Θ , Φ and Ψ gives the joint probability of corpus and latent topics given hyperparameters:

$$P(\mathbf{w}, \mathbf{z}|\alpha, \beta, \gamma) = \prod_{t=1}^T \frac{B(C_t^T + \beta)}{B(\beta)} \prod_{d=1}^D \frac{B(C_d^D + C_{d+1}^D + \alpha)}{B(\alpha)} \prod_{s=1}^S \frac{B(\gamma)}{B(C_s^S + \gamma)}, \tag{9}$$

where $B(\cdot)$ is the multinomial beta function, α , β and γ are vectors with const elements α , β and γ , respectively. C^T , C^D and C^S are three count matrixes. C_t^T , C_d^D and C_s^S are the t -, d - and s -th row of the matrixes C^T , C^D and C^S , respectively. The (t, w) -th element of C^T is the number of times that topic t is assigned to word w . The (d, t) -th element of C^D is the number of times that topic t is assigned to words in document d . The (s, t) -th element of C^S is the number of times that topic t is assigned to words in sequence s . Finally, the Gibbs sampling update for the topic z_n is obtained as follows

$$\begin{aligned}
 P(z_n = t | \mathbf{z}_{-n}, \mathbf{w}, \alpha, \beta, \gamma) &= \frac{P(z_n = t, \mathbf{w}, \mathbf{z}_{-n} | \alpha, \beta, \gamma)}{P(\mathbf{w}, \mathbf{z}_{-n} | \alpha, \beta, \gamma)} \\
 &\propto \frac{N_{-n,t}^{(w_n)} + \beta}{N_{-n,t}^{(\cdot)} + W\beta} \frac{N_t^{(d_n-1)} + N_{-n,t}^{(d_n)} + \alpha}{N^{(d_n-1)} + N_{-n}^{(d_n)} + T\alpha} \frac{N_{-n,t}^{d_n} + N_t^{(d_n+1)} + \alpha}{N_{-n}^{d_n} + N^{(d_n+1)} + T\alpha} \frac{N_{-n}^{(s_n)} + T\gamma}{N_{-n,t}^{(s_n)} + \gamma} \tag{10}
 \end{aligned}$$

where s_n is the sequence number of the word w_n , $N_{-n,t}^{(s_n)}$ is the number of times topic t occurring in the the facial image sequence s_n and $N_{-n}^{(s_n)}$ is the number of words in the sequence s_n (both excluding z_n). With a set of samples from the posterior distribution $P(\mathbf{z}|\mathbf{w})$, we can estimate Θ , Φ , and Ψ from \mathbf{w} and \mathbf{z} by equations

$$\hat{\theta}_{t|d} = \frac{N_t^{(d)} + N_t^{(d+1)} + \alpha}{N^{(d)} + N^{(d+1)} + T\alpha}, \hat{\phi}_{w|t} = \frac{N_t^{(w)} + \beta}{N_t^{(\cdot)} + W\beta}, \text{ and } \hat{\psi}_{t|s} = \frac{N_t^{(s)} + \gamma}{N^{(s)} + T\gamma}. \quad (11)$$

3.3 Applying TLTMs to Facial Expression Recognition

In facial expression recognition, TLTMs are learned for facial expression training dataset. The learned TLTMs for the i -th facial expression is denoted by a compact notation $\text{TLTM}^{[\text{Tr}^i]} = (\mathbf{w}^{[\text{Tr}^i]}, \mathbf{z}^{[\text{Tr}^i]}, \Theta^{[\text{Tr}^i]}, \Phi^{[\text{Tr}^i]}, \Psi^{[\text{Tr}^i]})$, here $\mathbf{w}^{[\text{Tr}^i]}$ is the image corpus of the i -th facial expression and $\mathbf{z}^{[\text{Tr}^i]}$ is the learned latent topic assignments. For a new facial image not contained in training dataset, we need to quickly assess the topic assignments, while the standard inference method described above is offline. Recently, some online [3] or efficient inference methods have been proposed [20], we adopt the efficient Monte Carlo algorithm as described in [20]. The basic idea of this method is to run only on the word tokens in the new image.

Given a testing facial image sequence $\{d_1^{[\text{Te}]}, d_2^{[\text{Te}]}, \dots, d_j^{[\text{Te}]}, \dots\}$, where the j -th image $d_j^{[\text{Te}]}$ has the bag-of-words representation $\{w_{j,1}^{[\text{Te}]}, w_{j,2}^{[\text{Te}]}, \dots, w_{j,52}^{[\text{Te}]}\}$. The trained TLTMs are used to classify the current image slice into one of the six basic expressions. Let $l_j^{[\text{Te}]}$ denote the label of the j -th testing image. Once the j -th image is obtained, we will sample new assignments of words to topics by applying equation (10) only to the word tokens in the j -th image. After several sampling iterations (20 iterations in our simulation), we can get the topic assignment $z_{j,k}^{[\text{Te}]}$ for each word in $d_j^{[\text{Te}]}$. The topic generation probabilities for image $d_j^{[\text{Te}]}$ in both document and sequence levels can be estimated by equation (11), and the probability $\theta_{t|d_j^{[\text{Te}]}, d_j^{[\text{Te}]}-1}^{[\text{Te}]}$ can thus be calculated by equation (7).

Finally, the observation probability of $d_j^{[\text{Te}]}$ conditioned on the i -th expression is calculated by

$$P(d_j^{[\text{Te}]}|l_j^{[\text{Te}]} = i) = \prod_{k=1}^{52} \sum_{t=1}^T \phi_{w_{j,k}^{[\text{Te}]}, t}^{[\text{Tr}^i]} \theta_{t|d_j^{[\text{Te}]}, d_j^{[\text{Te}]}-1}^{[\text{Te}]} \quad (12)$$

According to the Bayes's rule, the probability of sequence $\{d_1^{[\text{Te}]}, d_2^{[\text{Te}]}, \dots, d_j^{[\text{Te}]}\}$ classified to the i -th expression is calculated as

$$P(l_j^{[\text{Te}]} = i | \{d_1^{[\text{Te}]}, d_2^{[\text{Te}]}, \dots, d_j^{[\text{Te}]}\}) \propto \frac{1}{N(j)} P(d_j^{[\text{Te}]}|l_j^{[\text{Te}]} = i) \sum_{k=1}^6 P(l_j^{[\text{Te}]} = i | l_{(j-1)}^{[\text{Te}]} = k) P(l_{(j-1)}^{[\text{Te}]} = k | \{d_1^{[\text{Te}]}, d_2^{[\text{Te}]}, \dots, d_{(j-1)}^{[\text{Te}]}\}), \quad (13)$$

here $N(j)$ is a scale factor to ensure $\sum_{i=1}^6 P(l_j^{[\text{Te}]} = i | \{d_1^{[\text{Te}]}, d_2^{[\text{Te}]}, \dots, d_j^{[\text{Te}]}\}) = 1$ and $P(l_j^{[\text{Te}]} = i | l_{(j-1)}^{[\text{Te}]} = k)$ is the transition probability from expression k to i . To facilitate the computation of transition probabilities, a 6×6 matrix R is constructed. The (k, i) -th entry of R records the number of times transmitting from the expression k to i in two consecutive time slices. R is initialized to a matrix with all ones. The transition probability $P(l_j^{[\text{Te}]} = i | l_{(j-1)}^{[\text{Te}]} = k)$ is simply calculated as $R_i^k / \sum_i R_i^k$, where R_i^k is the (k, i) -th entry of the matrix S . All the probabilities involved in (13) are obtained, a testing facial image sequence is classified to expression i^*

$$i^* = \arg \max_{i=1, \dots, 6} P(l_j^{[\text{Te}]} = i | \{d_1^{[\text{Te}]}, d_2^{[\text{Te}]}, \dots, d_j^{[\text{Te}]}\}), \quad (14)$$

if $P(l_j^{[\text{Te}]} = i^* | \{d_1^{[\text{Te}]}, d_2^{[\text{Te}]}, \dots, d_j^{[\text{Te}]}\}) > 0.40$, otherwise Neutral expression is assigned.

4 Experiments

4.1 Dataset and Parameter Settings

We use the Cohn-Kanade Database to evaluate the performance of TLTM. This database consists of 100 university students ranging in age from 18 to 30 years. Sixty-five percent were female, fifteen percent were African-American and three percent Asian or Latino. For our experiments, we selected 72 whole image sequences (totally, 1085 images) from the database. Each expression contains 12 sequences. The original frames are normalized to 170×210 pixels facial images based on the positions of two eyes. Before using TLTM and LDA, we need to first set the hyperparameters α , β , γ and the number of latent topics T . For all runs of our algorithm, we set α , β and γ to constant values $\alpha = 50/T$, $\beta = 0.1$ and $\gamma = 60/T$. T is a very influential parameter for any latent topic models, and some Dirichlet Processes based methods have been proposed to estimate the value of T automatically [21]. In our simulation, we used the generally acceptable empirical methods to determine the optimal value for T . We run our model for different T values and found five latent topics provides the best recognition rate.

Table 1. The recognition rates (%) of LDA and TLTM

	JOY	SUR	ANG	DIS	SAD	FEA	Overall Rate
LDA	66.67	100.00	83.33	94.44	100.00	88.89	88.89
TLTM	75.00	100.00	91.67	100.00	100.00	95.83	93.75

4.2 Experimental Results

We used a three-fold cross validation in our experiments to verify the benefits of using TLTM to model facial expression dynamics. Table 1 presents the recognition results of LDA and TLTM. It can be observed that the TLTM method

outperforms the LDA method for the recognition of joy, anger, disgust and fear expressions, which confirms the benefit of using temporal information of image sequences. Furthermore, we can see that both methods perform relatively worse for the joy expression, since the joy expression mainly includes two AUs: AU6(Cheek raiser) + AU12(Lip corner puller) and the AU6 depends on some transient features such as nasolabial furrows presence and eye wrinkles increased, however AAM is not particularly suitable to track these features. Other tools (e.g. Canny edge operator) will be used to quantify the intensity of furrows and wrinkles in future work to obtain better performance.

Table 2. Comparisons with other methods

Methods	ParzenHMM	KnnHMM	DynamicLBP	SVMLBP	LDA	TLTM
Overall Rate	86.11	91.67	96.26	92.10	88.89	93.75

Table 2 summarizes a comparison to some other representative approaches. Here “ParzenWHMM” denotes a modified HMM in which the generation probability is estimated by a nonparametric density estimation method-Parzen Windows [10]. “KnnHMM” denotes a discriminate HMM proposed by Lefevre [14], in which the discrimination ability at hidden state level is improved by a k-nearest neighbors (k-NN) estimation method. “SVMLBP” denotes the method proposed by Shan [17], they used LBP to represent facial features and SVM as classifier. The “DynamicLBP” method used dynamic LBP to represent facial features and used SVM as classifier [27]. It can be observed that the LDA method performs better than the HMM based method and slightly worse than its discriminant version KnnHMM. Our method achieves the similar performance as the SVMLBP method, although TLTM is a generative model and does not use the information of other classes in the training stage. The main difference of the methods DynamicLBP and SVMLBP is LBP is replaced by dynamic LBP, which confirms the benefit of considering the temporal information for sequential data classification. The DynamicLBP method performs better than our generative model, since SVM is a discriminant model which uses the information of other classes in the training stage. Recently, some works on increasing the discriminant ability of LDA have been proposed such as DisLDA [13] and MedLDA [28]. We will use some discriminant rules to train our TLTM in future work to get higher recognition rate.

4.3 Some Recognition Examples

In this section, we will use two examples to illustrate the efficiency of the proposed method in an intuitive way. In the first example, we created a short image sequence as shown in Figure 5(a) in which the subject performed smiling with blinking her eyes in the frames 4 and 5. We can observe that from the second frame lip corners begin to be pulled obliquely and cheeks are raised. From Fig. 5(b), we can see the probabilities of the six expressions are close in the first

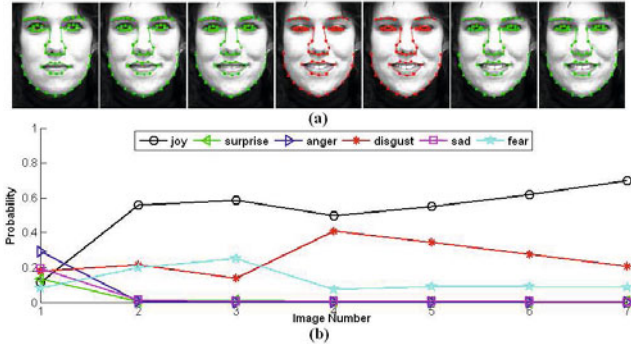


Fig. 5. Example 1: (a) An image sequence shows a subject performing smiling with blinking eyes in the frames 4 and 5, (b) the probability distributions of facial expressions

frame. As the expression progresses with time the probability of joy increases gradually and decreases in the frames 4 and 5 resulted by the eyes blinking action. In the 7-th frame, the probability of joy rises to nearly 0.7 and implies that this frame has the apex joy expression. This experiment illustrates that our method can well model the evolution of facial expression.

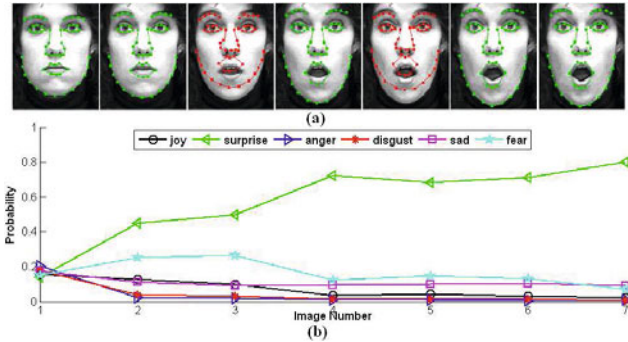


Fig. 6. Example 2: (a) An image sequence shows a subject performing surprise with tracking error in the frames 3 and 5, (b) the probability distributions of facial expressions

Figure 6(a) shows another image sequence in which the subject performed surprise with some frames mis-tracked. In frames 3 and 5, we can see that the locations of mouth and chin are tracked in error. Fig. 6(b) gives the result of our method, from which we can observe that although the probability of surprise visibly decreases in the 5-th frame, the facial expression can still be correctly recognized. This example illustrates that our method is robust to tracking error.

5 Conclusions and Future Work

This paper proposed a new latent topic model TLTM for facial expression analysis by integrating the temporal information of image sequences. We redefined the topic generation probability without involving new latent variables or increasing inference difficulties. Experiments on CMU expression database confirmed the efficiency of the TLTM in facial expression recognition. In future work, we will pay more attention to feature extraction and use some discriminant training rules to increase the discriminant ability of TLTM to get better performance.

References

1. Bassili, J.: Emotion recognition: the role of facial movement and the relative importance of upper and lower areas of the face. *Personality and Social Psychology*, 2049–2059 (1979)
2. Blei, D., Ng, A., Jordan, M.: Latent dirichlet allocation. *JMLR* 3(2-3), 993–1022 (2003)
3. Canini, K.R., Shi, L., Griffiths, T.L.: Online inference of topics with latent Dirichlet allocation. In: *AISTATS* (2009)
4. Chang, Y., Hu, C., Turk, M.: Probabilistic expression analysis on manifolds. In: *CVPR*, pp. 520–527 (2004)
5. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. *IEEE Trans. on PAMI* 23(6), 681–685 (2001)
6. Ekman, P., Friesen, W.V.: *Facial Action Coding System (FACS): Manual*. Consulting Psychologists Press, Palo Alto (1978)
7. Griffiths, T.L., Steyvers, M.: Finding scientific topics. *PNAS* 101, 5228–5235 (2004)
8. Hanna, M.W.: Topic modeling: beyond bag-of-words. In: *ICML* (2006)
9. Hospedales, T., Gong, S., Xiang, T.: A Markov clustering topic model for mining behaviour in video. In: *ICCV* (2009)
10. Jin, N., Mokhtarian, F.: A non-parametric HMM learning method for shape dynamics with application to human motion recognition. In: *ICPR*, pp. 29–32 (2006)
11. Kanade, T., Cohn, J.F., Tian, Y.: Comprehensive database for facial expression analysis. In: *FG*, pp. 46–53 (2000)
12. Kumano, S., Otsuka, K., Yamato, J., Maeda, E., Sato, Y.: Pose-invariant facial expression recognition using variable-intensity templates. In: Yagi, Y., Kang, S.B., Kweon, I.S., Zha, H. (eds.) *ACCV 2007, Part I. LNCS*, vol. 4843, pp. 324–334. Springer, Heidelberg (2007)
13. Lacoste-Julien, S., Sha, F., Jordan, M.I.: DiscLDA: Discriminative learning for dimensionality reduction and classification. In: *NIPS*, pp. 897–904 (2008)
14. Lefevre, F.: Nonparametric probability estimation for HMM-based automatic speech recognition. *Computer Speech and Language* 17(2-3), 113–136 (2003)
15. Fei-Fei, L., Perona, P.: A bayesian hierarchical model for learning natural scene categories. In: *CVPR*, pp. 524–531 (2005)
16. Minka, T., Lafferty, J.: Expectation propagation for the generative aspect model. In: *UAI*, pp. 352–359 (2002)
17. Shan, C., Gong, S., McOwan, P.W.: Robust facial expression recognition using local binary patterns. In: *ICIP*, pp. 370–373 (2005)
18. Shang, L., Chan, K.P.: Temporal Exemplar-Based Bayesian Networks for Facial Expression Recognition. In: *ICMLA*, pp. 16–22 (2008)

19. Shang, L., Chan, K.P.: Nonparametric Discriminant HMM and Application to Facial Expression Recognition. In: CVPR, pp. 2090–2096 (2009)
20. Steyvers, M., Smyth, P., Rosen-Zvi, M., Griffiths, T.: Probabilistic Author-Topic Models for Information Discovery. In: KDD, pp. 306–315 (2004)
21. Teh, Y.W., Jordan, M.I., Beal, M.J., Blei, D.: Sharing Clusters among Related Groups: Hierarchical Dirichlet Processes. In: NIPS (2004)
22. Tian, Y., Kanade, T., Cohn, J.: Recognizing action units for facial expression analysis. *IEEE Trans. on PAMI* 23(2), 97–115 (2001)
23. Wang, X., Grimson, E.: Spatial latent dirichlet allocation. In: NIPS (2007)
24. Yang, P., Liu, Q., Metaxas, D.N.: Boosting coded dynamic features for facial action units and facial expression recognition. In: CVPR, pp. 1–6 (2007)
25. Yeasin, M., Bullot, B., Sharma, R.: From facial expression to level of interest: a spatio-temporal approach. In: CVPR, pp. 922–927 (2004)
26. Zhang, Y., Ji, Q.: Active and dynamic information fusion for facial expression understanding from image sequences. *IEEE Trans. on PAMI* 27(5), 699–714 (2005)
27. Zhao, G., Pietikäinen, M.: Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Trans. on PAMI* 29(6), 915–928 (2007)
28. Zhu, J., Ahmed, A., Xing, E.P.: MedLDA: Maximum margin supervised topic models for regression and classification. In: ICML (2009)

From Local Features to Global Shape Constraints: Heterogeneous Matching Scheme for Recognizing Objects under Serious Background Clutter

Martin Klinkigt and Koichi Kise

Graduate School of Engineering, Osaka Prefecture University

Abstract. Object recognition in computer vision is the task to categorize images based on their content. With the absence of background clutter in images high recognition performance can be achieved. In this paper we show how the recognition performance is improved even with a high impact of background clutter and without additional information about the image. For this task we segment the image into patches and learn a geometric structure of the object. In evaluations we first show that our system is of comparable performance to other state-of-the-art system and that for a difficult dataset the recognition performance is improved by 13.31%.

1 Introduction

To categorize images is a long researched task in computer vision which belongs to the field of image recognition. Recently this field is of high interest and the results are remarkable. In the more challenging object recognition only a part of the image is showing the object of interest, while the other would show unrelated object which we name background. If the amount of information extracted from the background overweight the amount of useful information extracted from the object, the results of object recognition can become unreliable. To avoid this problem the image is often not described in a global manner. By only looking like through a peephole the information in the image is described in tiny uncorrelated parts so-called local features. The drawback of this approach is that it lose discriminative power which would be needed to distinguish between similar objects.

For the gain of more discriminative power even for images with background clutter, researchers often apply pragmatic solutions, e.g., segment the object in the image from the background [1] or apply clustering [2] to learn frequently used visual words. However, also these approaches have some drawbacks which make their application difficult. A segmentation of the object can hardly be done automatically and thus must be provided by human. With the help of clustering it can be distinguished between information of the object and the background. This approach relies on a large number of images, which can only be provided in some impractical contexts.

One possible way to increase the discriminative power is the weak geometric consistency as proposed by Jegou et al. [3] which verifies some global geometric information. However, it can address the problem of background clutter only to a limited extend, e.g., the image must mainly show the object.

By understanding the nature of background clutter we go stepwise from local features to more powerful global description. In that way we can even work with only one training image per object. In such a case it is normally not clear which parts of the image belong to the object and which to the background. To address these problems, we assume that information from the background is mainly unstructured concerning the shape which involves two ideas. First, smaller patches of the image can only belong either to the object or to the background which holds for all reasonable use cases. Second, during recognition it is possible to distinguish between object and background with a shape model. Matching information with a consistent and structured shape is a hint for the object, while unstructured matching could be background clutter and will be ignored. By going from these small patches to larger areas, with a novel use of a shape context and finally defining a overall global shape with the reference point, we can even handle occlusions or distinguish between similar looking objects. In an evaluation on a challenging dataset we compare our system to an approach at only works on local features and the model of the weak geometric consistency. Our system achieves an increased recognition performance of 13.31%.

2 Related Work

Our motivation is to improve recognition performance for images containing background clutter by keeping the additional work for the user low. Concerning this objective only a few has been done. Since we only work with the information of the images itself, the environment is the same as in the context of similar image search. A representative of such a task is provided by Jegou et al. [4] which works with local features. Jegou et al. use a sophisticated Hamming embedding to search within a large image database. This system was not designed to address the problem of background clutter and could be easily disturbed. This due to the fact that it was designed for image rather than object recognition.

In recent research scientists have put some interest on the configuration of features to use more discriminative global information. All proposed approaches can not fully address the problem of background clutter. The weak geometric consistency (WGC) [3] is such an approach. In this model the system makes use of the scale and orientation of the Scale-Invariant Feature Transform (SIFT) proposed by Lowe [5]. The idea of the WGC is that the features of the object are transformed consistently, and therefore, beside some noise these changes in the shape are for all features the same. However, this holds only if the system has not to struggle with much background clutter. The used global histograms, which describe the whole image at once, are easily disturbed by features from the background which will finally become meaningless.

By working on pre-segmented image patches Plath et al. [6] improve object detection and its segmentation. The two major differences to our proposed method

are that the system of Plath et al. is designed to work on large learning datasets for the objects and a bounding box is used to limit the effect of the background.

The use of a shape context to improve recognition performance in images was also proposed by Mortensen et al. in [7]. In their approach maximum curvature at each pixel is stored in the shape context. Such a curvature is not robust against background clutter and can hardly be improved to be so. Since it can not be distinguished between curvature from the background and the object, curvature from background will suppress curvature from object, if their amount becomes too large. In contrast we use geometric information from more robust PCA-SIFT features as provided by Rahul et al. [8].

The implicit shape model as proposed by Leibe et al. [9] express the shape of features concerning one selected reference point. The system of Leibe et al. requires several hundreds of images with a bounding box around the object to limit the effect of background clutter. This highly engrosses the user who has to provide this data. In our utilization of such a reference point we create the shape model during recognition and learn images without a bounding box. Compared to previous work [10], we work on the pre-filtered features to reject irrelevant information.

3 Proposed Method

The source of the background clutter problem is erroneous matching of local features. To identify such erroneous matches it makes sense to take into account the configuration of these local features. However, this lead to the new problem of defining a suited model. Such a model must be strong enough to detect erroneous matches and on the other hand it can not be too strict to endure occlusions of the object of interest. Facing these problems the use of only one model easily fails to achieve both.

In our proposed method we go stepwise from local features to a description of the local area around the feature and finally to a more global shape. Figure 1 gives an overview of our system. From the training images we extract PCA-SIFT features [8]. These features are frequently used in computer vision and good results have been achieved. The features together with some information about their location in the image are stored in a database.

In the first step a reduction of erroneous matching is achieved by working on smaller patches of the image. The same type of local features are extracted and compared with features stored in the database. The reason for the use of these local features are their stability. Even for changing lighting conditions and blurred image, these features could be extracted reliable. For these patches we assume that they can only belong either to the object or to background. Therefore, patches with a even evidence for many different objects are ignored.

For the remaining patches the local configuration of the features is verified. This is done by the calculation of a local shape context from the features. In small regions changes in the configuration due to noise, perspective changes etc. are only little. A strict model as the shape context is suited for such a task. To be fault-tolerant concerning occlusion and to separate between similar objects,

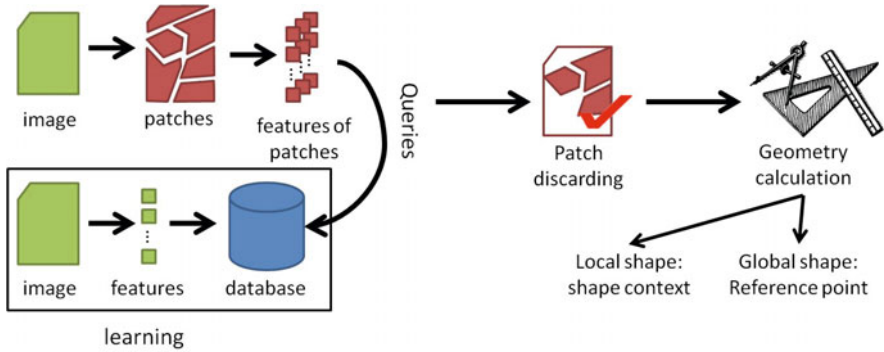


Fig. 1. System overview

we use an additional model by utilizing a reference point. This is needed, since different objects can have a high similarity even in relatively large parts, e.g., windows or doors of buildings. Concerning the global shape such differences can be detected.

This section is organized as follows: First we explain the voting scheme in section 3.1 followed by the description about the image patches and how we discard them in section 3.2. After that give the details about the shape context in section 3.3 and the reference point in section 3.4. At the end in section 3.5 we describe how we use this information to score for the object.

3.1 Voting Scheme in Object Recognition

Voting is often used to recognize objects, since its implementation is simple and much faster compared to complex vocabulary learning. For voting local features are used to represent an image and, therefore, an object. These features together with the object ID are stored in a database. For each feature extracted from the query image similar feature in the database is searched. If features are found which fulfill a certain threshold in similarity, the match cast a vote for the corresponding object. The object with the most votes is supposed to be the correct result.

3.2 Image Patches

By processing the whole image at once including all background clutter, it can not be distinguished between the background and the object. Instead we work on smaller pre-segmented image patches for which we assumed that they can only belong either to the object or to the background. These small patches can be discarded, if there is no clear evidence for one object. This assumption holds for most of the realistic use cases. A simple solution for the task of segmentation is to take the color information. Our implementation is based on the graph-based segmentation as provided by Felzenszwalb et al. [11]. Figure 2 shows such a segmentation for a real image. On the left side (Fig. 2a) we see the original

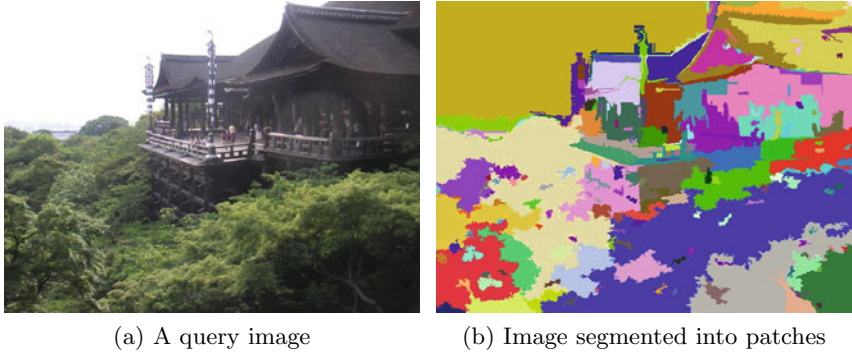


Fig. 2. Example segmentation of a query image

image and on the right side (Fig. 2b) the individual colored patches after the application of the segmentation algorithm. We can see the object is segmented into many different patches. However, problems do not result from this over segmentation and our assumption still holds.

For each patch we decide which object it is showing. Here we only use the PCA-SIFT features and the voting scheme described in section 3.1. The system returns a ranked list with confidence values z_{SP_o} for the objects $o \in O$ by counting the number of good matching features of o . If the patch cannot be assigned to a few objects with a high confidence, the whole patch is discarded. Here we just apply a simple threshold based on the number n of objects which have some similarity with the patch and n_{top} be the number of objects with high confidence. The patch will be used only if $n_{top} = 1$ or $n_{top}/n < 3/4$. If a patch has similarity to too many different objects, it cannot be used for object recognition, since information from this area is not reliable.

3.3 Shape Context

After ambiguous patches have been discarded based on the PCA-SIFT features we verify the local configuration of the features in the remaining patches. First we utilize a shape context for this task. We take the position of the features and ignore from which patches they are extracted. This is due to the fact that the object maybe was segmented into several patches, as we can see in Fig. 2. With this approach we build a strict model of the near surrounding of a feature.

The shape context was introduced by Belongie et al. in [12]. For a certain point a log-polar histogram is calculated. This histogram describes the number of points from the border of the object in a small region. Figure 3a indicate such a shape context with its segmentation of the surrounding area into regions. To achieve a proper recognition the orientation γ_{SC} and scale d_{SC} of the shape context must be the same during training and recognition. Otherwise the content of the areas are not comparable. By using SIFT features we can ensure these conditions by making use of the scale l and the orientation θ . These properties

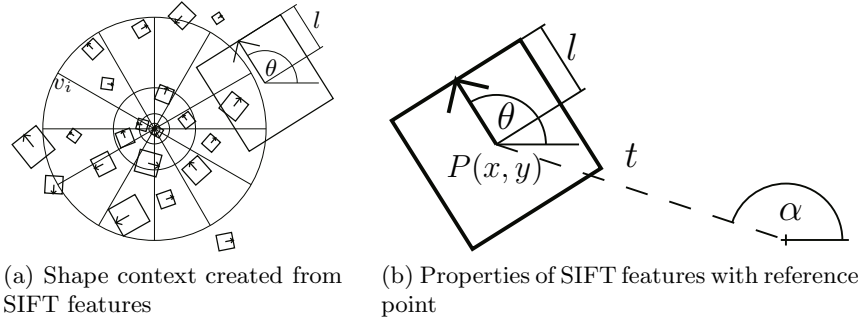


Fig. 3. Illustration of used local configuration information

are adjusted to the size and orientation of the object in the image and illustrated in Fig. 3a.

Let $m_i = (l_d, \theta_d, l_q, \theta_q) \in M$ the set of matching features between the query and the database image. Let l_d be the scale of the feature from the database, θ_d is its orientation and let respective l_q and θ_q are these values for the feature extracted from the query image.

Calculation of a Shape Context. For the calculation of one shape context we place it over the PCA-SIFT feature which we call the center feature c of the shape context. We adapt the orientation γ_{SC} of the shape context to be same as the orientation θ of the center feature c and the diameter d_{SC} of the shape context is adapted relatively to the scale l of the center feature c . The shape context now segments the surrounding area into smaller regions $v_i \in V$ as illustrated in Fig. 3a. For the other matching features we analyse in which certain region v_i they are laying. The costs for this are minor, since we only have to store some pointers to the already stored features.

For the parameters of the shape context we choosed the following setting: 4 sectors each 90° for the orientation, a default starting size of 200 pixels of the diameter γ_{SC} multiplied with the scale l of the feature separated into 2 regions.

Verification of a Shape Context. After we made the first filtering of ambiguous patches we first analyse whether the matching features from the database and the query image lay in the same region of the shape context. If the matched features are in different regions, then this is a hint that the shape is teared to pieces. These features are removed from the shape context.

One of the major difficulties in using a shape context is the definition of a distance or similarity function. Our approach is inspired by the weak geometric consistency. Jegou et al. claimed that features from the object can only be transformed consistently. We make the same simplifications to calculate a score of similarity. The differences in the scale l and orientation θ of the features are quantized and stored in histograms δ_l and δ_θ . Let $g(\delta_{l_j})$ be the score in scale difference histogram bin δ_{l_j} and respectively $h(\delta_{\theta_j})$ the score in a bin of orientation difference histogram. The score s resulting from the matches $m_i \in M$ is:

$$s = \min \left(\max_{\delta_{l_j} \in \delta_l} g(\delta_{l_j}), \max_{\delta_{\theta_j} \in \delta_\theta} h(\delta_{\theta_j}) \right). \quad (1)$$

The sum of the scores s of all shape context is the final confidence value z_{SC_o} for a certain object $o \in O$ based on the shape context.

Since the shape context is so strict, the number of false positives is really low with the drawback that occlusions or perspective changes can only be addressed to a limited extend. Also if the shape context is used to describe the global shape it will run into the same problems as the weak geometric consistency. To benefit from information about the global shape and have a more flexible model to address the problem of occlusions we use the reference point.

3.4 Reference Point

The shape context is a strict model and can become fragile like in the case of occlusion. With a global model such a problem can be addressed. However, a proper cleaning from features with low evidence is needed to ensure that is created for the object which we have done in section 3.2 by discarding ambiguous regions. For the global shape we choose a fixed reference point approach. Concerning this reference point the location of all features in the image is expressed. During recognition we create dynamically our reference point RP from the remaining features of the database image. The local configuration of the features in the query image is verified with the help of this RP . In this section we only give a short explanation. For more details we refer to previous work [10].

Learning. In the learning phase of such a reference point the additional effort is low. Beside storing the position $P = (P_x, P_y)$, the scale l and orientation θ of the features nothing must be done. The main calculation is done during verification.

Verification. Let M be as before our set of matched features cleared from features of ambiguous patches. The reference point is created from the features of the database image. Therefore, let F_D be the set of features in the database image and $P = (P_x, P_y)$ the position in the image of such a feature from F_D . We select the centroid of all features in F_D as position of the reference point $RP = (RP_x, RP_y)$. This reference point RP is used to obtain two new values. These are the distance t of the feature to this reference point RP and the enclosed angle α which are shown in Fig. 3b. The feature attached with these properties is then placed over the corresponding feature from the query image. After we adapted the scale and the orientation of P to be equal to the values of the feature from the query image, we get a proposal for the position of the reference point in the query image. These steps are done for all matches. Dense regions of reference point proposals indicate well fitting feature configurations, while sparse regions would give a hint that the shape is again teared to pieces. Only dense regions are used to vote for the object. Sparse regions are ignored. Taking the number of features leading to dense agglomerations of reference points, gives the confidence value z_{RP_o} for a certain object $o \in O$.

3.5 Final Scoring

For the final score any balancing between the different approaches can be implemented. We found such a balance with the help of the machine learning approach learning-to-rank. We took into account the patch score z_{SP_o} of useful patches, the shape context score z_{SC_o} and the score z_{RP_o} from the reference point approach of an object $o \in O$. The final score z_o for the object o is:

$$z_o = \alpha z_{SP_o} + \beta z_{SC_o} + \gamma z_{RP_o} \quad (2)$$

with the weights $\alpha = 0.45$, $\beta = 0.35$ and $\gamma = 0.2$. These weights were determined based on preliminary experiments. We can see that the image patches have the highest contribution to the final score. Concerning the both shape models, the shape context plays a more important role as compared to the reference point. As we pointed out in section 3.3 the shape context is a strict model resulting in a small number of false positives.

4 Experiments

We performed evaluations on three different datasets. First we used the publicly available N-S dataset [13] and INRIA holiday dataset [14]. With these evaluations we determined the performance of ours to other state-of-the-art systems. The last evaluation was done on our own dataset, which is more challenging since it includes a higher impact of background clutter. As performance measurement we used the mean average precision (mAP) [15] which combines precision and recall for a ranked list of results. Higher values indicate better recognition performance of the system.

4.1 Comparison Datasets

The N-S dataset consists of 4 images for each of the 2550 objects. Hence this dataset contains 10200 images. In this evaluation all images are used as database and query images at the same time. Ideally the system returns the relevant 4 images at the first 4 ranks in a ranked list. A large number of the objects was taken in front of exactly the same structured background (carpet), which is one of the most challenging problem for our system on this dataset. The feature descriptors and the shapes become almost the same for many different objects. However, our achieved mAP is 61.58% or a score of 2.5 in the terms of Nistér et al. and is comparable to the results from smaller quantizer reported on [16].

INRIA’s holiday dataset [14] consists of 1491 images separated into 991 training and 500 query images. For the query images the system should again rank the correct images at the top position. On this dataset we achieve a mAP of 71.83% and again this is comparable to 75.07% which was reported by the authors on this dataset.

At this point we note that we achieved these scores only with simple voting based on the PCA-SIFT features. The use of image patches or additional shape



Fig. 4. Example query images from the temple data set. From left to right 1 images of Ginkaku-ji, 2 images of Kinkaku-ji and 1 image of Kiyomizu-dera are shown.

information cannot improve these results significantly, since these datasets are not well suited for our objective to handle background clutter. We discuss about this in the next section. Also we mention that up to now we have not applied any affine region detector nor time consuming clustering and vocabulary training, as the authors in [13] and [3] did. So we left some space for further improvements on these datasets.

4.2 Temple Dataset

The evaluations on the N-S dataset and INRIA’s holiday dataset show that our system is of equal performance even without vocabulary learning or clustering. These datasets are somehow simple for our use case. On one hand they sometimes include large affine transformations and rotations and on the other hand only moderate changes in scale. Hence, the impact of background clutter is limited. These datasets are suited for image recognition where the challenge against background clutter is not the major objective. Therefore, we created our own dataset which contains only a few “good” images showing the object nearly perfectly. This dataset consists of various images of temples and shrines in Japan. Due to the nature of such scenery images as a normal tourist would take them, they contain many side objects, e.g., persons or trees. Figure 4 shows a short sequence of these images. All 107 images are used as queries and the system returns a ranked list of objects rather than images as it was the case in the previous evaluations. As the training dataset we did not prepare any images. We used all images provided by Wikipedia for these buildings. In detail we learned all buildings, which belong to the classes “temple in Kyoto Prefecture” and “treasure of Japan”. The number of objects is 84 while the number of provided images is 819. So on average the system has less than 10 images per object. By taking all instances of these classes, the objects among which the system has to distinguish, look quite similar.

This construction of the evaluation is suited for our objective. The objects have only minor differences and the impact of background clutter is huge (compare the first and the last image of Fig 4). There might be some connection of the building with its background, since they appear together with trees. However, these trees are shown together with all objects and still we have to struggle with natural changes like different seasons (red autumn leaves or snow covering the building).

Table 1. Results for the temple dataset. Shown is the mean average precision (mAP) in percentage.. For column (1) we set the weight $\beta = 0$ of the shape context in equation 2. Column (2) shows the result for our proposed method with equal weights α, β, γ . The last give the best results for any possible combination of the WGC with the other scores.

	single score results					combined score results		
	SIFT	WGC	SP	SC	RP	(1)	(2)	(3)
Ginkaku-ji	24.72	29.87	28.87	22.60	31.20	31.68	41.92	37.67
Kinkaku-ji	45.36	44.02	43.83	33.91	23.41	50.71	52.05	49.73
Kiyomizu-dera	15.87	26.47	18.25	23.41	21.37	22.97	31.92	25.94

Again we use the mAP as our performance measurement. As we constructed our use case in the manner of object recognition only one result is correct. A low rank leads directly to a dramatically decreasing mAP. The results are shown in Table 1 split for the different objects and scoring functions. For further analysing of the behavior on increasing databases, we loaded in a similar manner as Jegou et al. [3] 100,000 images from Flickr. These images have no relation to any object and are just added to disturb the system. The results of experiments with distractor images are shown in Fig 5.

We used the following abbreviations: simple SIFT matching (SIFT), segmentation into image patches (SP), weak geometric consistency (WGC), shape context approach (SC) and reference point voting (RP). As we can see from the results in Table 1 none of the approaches alone has the overall best performance for all objects. From this we can conclude that for the objects different type of information is important. For Kinkaku-ji this seems to be the PCA-SIFT features. The reason for this may be that this object can be well recognized concerning small details. When we look at the third image of Fig. 4 we notice that only a small part of the object is visible. This small part is enough to recognize the object. For the other two objects we can conclude that the shape information plays an important role. These objects have only less characteristic features since both of them have the same type of roof and are constructed from wood.

For the results showing in column (1) we set $\beta = 0$ in equation 2. So only the results from the image patches and the reference point are used. By using only these two scores the recognition performance is improved for all objects. In the case of our full proposed method shown in column (2), the results for all objects are improved significantly. For the results of column (3) we modified equation 2 to make use of the WGC with global histograms. Shown are the best achieved results and as we can see our proposed method with the shape context is superior.

Working on a larger database our proposed method can keep the good performance with a mAP of around 10% even up to 100k distractor images as shown in Fig. 5. In an interval from 10k to 30k distractor images the approach of images patches outperforms all other approaches. For the WGC we can see that it can only achieve an improvement in recognition performance for a database without distractor images. As soon as distractor images are included in the database, the performance drops significantly.

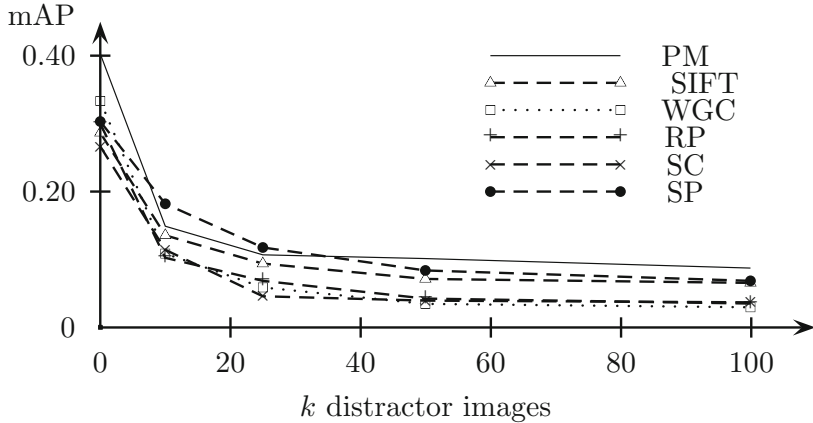


Fig. 5. Mean average precision for temple dataset with increasing number of distractor images. Shown are the graphs for the single scores (SIFT, WGC, SC, RP) and our proposed method (PM).

5 Conclusion

In this paper we have discussed object recognition systems and the resulting problems of background clutter. We have addressed this problem by keeping the additional effort at training time low. During recognition we work on image patches and use information about the local configuration of features.

With two evaluations on publicly available datasets we have shown that our system is of comparable performance by refrain from applying time consuming tasks like affine region detectors or clustering to obtain visual words. Working on our own challenging dataset with a high impact of background clutter, we achieved an improvement of 13.31% in terms of mean average precision. The results were achieved without a segmentation of the object from the background of the image.

Further research will concentrate on an object dependent combination of the shape information and how this information can be used to localize the object in the image.

Acknowledgment. This work was supported in part by the Grant-in-Aid for Scientific Research (B)(19300062) from Japan Society for the Promotion of Science(JSPS).

References

1. Opelt, A., Pinz, A., Zisserman, A.: A boundary-fragment-model for object detection. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 575–588. Springer, Heidelberg (2006)
2. Tirilly, P., Claveau, V., Gros, P.: Language modeling for bag-of-visual words image categorization. In: Proc. of the International Conference on Content-Based Image and Video Retrieval, CIVR, pp. 249–258. ACM, New York (2008)

3. Jégou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 304–317. Springer, Heidelberg (2008)
4. Jégou, H., Douze, M., Schmid, C.: Packing bag-of-features. In: Proc. of ICCV (2009)
5. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proc. of ICCV, p. 1150 (1999)
6. Plath, N., Toussaint, M., Nakajima, S.: Multi-class image segmentation using conditional random fields and global classification. In: Proc. of ICML, pp. 817–824. ACM, New York (2009)
7. Mortensen, E.N., Deng, H., Shapiro, L.: A sift descriptor with global context. In: CVPR (2005)
8. Rahul, Y.K., Ke, Y., Sukthankar, R.: Pca-sift: A more distinctive representation for local image descriptors. In: Proc. of IEEE CVPR, pp. 506–513 (2004)
9. Leibe, B., Leonardis, A., Schiele, B.: Combined object categorization and segmentation with an implicit shape model. In: ECCV Workshop on Statistical Learning in Computer Vision, pp. 17–32 (2004)
10. Klinkigt, M., Kise, K.: Using a reference point for local configuration of sift-like features. In: Meeting on Image Recognition and Understanding, MIRU (2010)
11. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. *Int. J. Comput. Vision* 59, 167–181 (2004)
12. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. *IEEE PAMI* 24, 509–522 (2002)
13. Nistér, D., Stewénius, H.: Scalable recognition with a vocabulary tree. In: CVPR, vol. 2, pp. 2161–2168 (2006)
14. Jégou, H., Douze, M.: Inria holidays dataset (2008), <http://lear.inrialpes.fr/people/jegou/data.php>
15. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: Proc. of CVPR (2007)
16. Nistér, D., Stewénius, H.: Recognition benchmark images (2006), <http://vis.uky.edu/%7stewe/ukbench/>

3D Structure Refinement of Nonrigid Surfaces through Efficient Image Alignment

Yinqiang Zheng, Shigeki Sugimoto, and Masatoshi Okutomi

Department of Mechanical and Control Engineering, Tokyo Institute of Technology

Abstract. Given a template image with known 3D structure, we show how to refine the rough reconstruction of nonrigid surfaces from existing feature-based methods through efficient direct image alignment. Under the mild assumption that the barycentric coordinates of each 3D point on the surface keep constant, we prove that the template and the input image are correlated by piecewise homography, based on which a direct Lucas-Kanade image alignment method is proposed to iteratively recover an inextensible surface even with poor texture and sharp creases. To accelerate the direct Lucas-Kanade method, an equivalent but much more efficient method is proposed as well, in which the most time-consuming part of the Hessian can be pre-computed as a result of combining additive and inverse compositional expressions. Sufficient experiments on both synthetic and real images demonstrate the accuracy and efficiency of our proposed methods.

1 Introduction

3D recovery of non-rigid surfaces from individual images is still a challenging task in computer vision due to its inherent ambiguity, which requires taking full advantage of available image information and other proper constraints to disambiguate the reconstruction. Such additional constraints range from physical knowledge in physics-based 3D recovery, e.g. [1,2] among many others, to temporal consistency in 3D tracking [3,4] and template-free recovery [5], and to geometric constraints in non-rigid 3D detection [6,7,8]. In this paper, we consider inextensible non-rigid surfaces and incorporate the constraints on the surface mesh edges as in [6,8]. Our concentration is on the usage of the surface texture so as to handle sparsely textured nonrigid surfaces with sharp shape details, such as creases and folds as shown in Fig.1. Many other image cues, like silhouettes and contours ([1,9,10] to cite a few), have also been used for non-rigid 3D recovery, but we do not consider them here.

According to how to make use of surface texture, the majority of existing methods for non-rigid 3D recovery can be roughly categorized into two groups:

Feature-based methods: The feature-based methods establish 3D-2D feature correspondences in template-based recovery [3,4,6,7,8], or 2D-2D ones for a long video sequence in Deformable Structure from Motion [11,12] or simply for two consecutive frames [5], and then recover 3D structure by minimizing, explicitly or implicitly, certain measurement of reprojection error. The objective function is

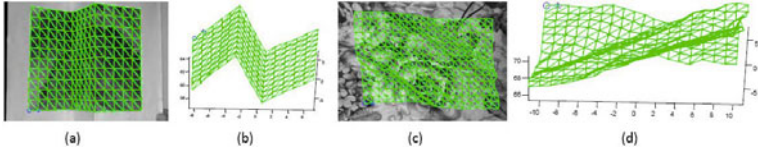


Fig. 1. Reconstruction of inextensible surfaces from single images. Existing feature-based methods tend to oversmooth the sparsely textured and sharply creased surface in (a); For a relatively well-textured surface with sharp folds (c), our image alignment method can improve its photo-consistency. (b) and (d) are their corresponding 3D meshes, respectively.

relatively easy to optimize, thus best suited to detect shape from a single image. However, to obtain a reliable reconstruction, the surface should be well-textured with dense salient features over the whole surface, which is generally not the case in practice. To tackle poorly textured surfaces, some prior cues, like the smoothness assumption [13] and the deformation model [6, 7, 8], are frequently introduced, whose results are roughly correct but weak in photo-consistency, like Fig.1(c). More seriously, such prior knowledge tends to oversmooth sharp details, thus can not be used to accurately recover sharply creased surfaces. Another possible alternative is to establish dense correspondences through non-rigid 2D registration as in [3, 6]. However, even the state-of-the-art methods [14, 15] introduce some shape terms to penalize sharp deformation, thereby inapplicable for sharp creased and folded surfaces.

Appearance-based methods: The appearance-based methods (or direct methods) take advantage of the intensity of each pixel of the surface image, and consequently are able to obtain photo-consistent reconstruction even for poorly textured surfaces. Unfortunately, the resulting objective function is highly non-convex, thereby commonly used in a tracking scenario. Another well-known drawback of the appearance-based methods is its inefficiency due to re-evaluation of the Hessian for each pixel at each iteration. Interestingly, some efficient Inverse Compositional Image Alignment (ICIA) algorithms have been proposed for 3D tracking of human face [16]. To the best knowledge of the authors, no such efficient algorithm exists for generic non-rigid surfaces.

In order to accurately reconstruct (or more exactly detect) the 3D structure of sparsely textured surfaces from a single image, it is desirable to fuse the feature-based and appearance-based methods, so that we can initialize the non-convex image alignment by using the easy-to-solve feature-based methods and derive photo-consistent shape from each pixel of the surface image, rather than from the prior knowledge. Such fusion has been proved feasible for fast non-rigid 2D recovery [14]. In this paper, we extend it to 3D case.

Given a template image whose 3D structure is known, we follow the feature-based robust convex method combined with local deformation model [8] and the closed-form solution with global deformation model [6] to derive rough reconstructions, which are used to initialize the iterative appearance-based methods.

Under the mild assumption that the barycentric coordinates of each point on its corresponding patch of the 3D surface are constant, a basic assumption underlying many non-rigid 3D surface recovery methods [3, 4, 6, 7, 8], we show that the template image and the input image are correlated by piecewise homography. Based on this homography warp, we propose a direct Lucas-Kanade method, also known as Forward Additive Image Alignment method [17], to recover sparsely textured surfaces by integrating the constraints on mesh edges to disambiguate the reconstruction. Since we never introduce the smooth term or the deformation model at this step, the direct image alignment method enables us to tackle sharp details with strong photo-consistency. The well-known downside of the direct Lucas-Kanade image alignment lies in its inefficiency, since the Jacobian and Hessian should be recomputed for each pixel at each iteration. To improve efficiency, an equivalent method is proposed as well by combing the additive and inverse compositional expressions. Although the Hessian is not completely constant across iteration, the most time-consuming part can be computed offline, while the inconstant part needs only to be recomputed for each patch of the surface, not for each pixel on the input image. This makes it much faster than the direct Lucas-Kanade method, although the folds of acceleration are dependent on the resolution of the surface mesh. In a typical experiment with a 11x16 triangulated mesh and 720x576 images, it takes about 0.2s to compute the Hessian, in contrast to 2.8s in the direct Lucas-Kanade method.

In the remaining of the paper, we first derive the warp function between the template and the input image in section 2, based on which the appearance-based methods for non-rigid 3D recovery are introduced in section 3. We present the fusion method in Section 4. Section 5 shows the extensive experimental results for both synthetic and real images and section 6 includes some concluding remarks of this paper.

2 Warp between the Template and the Input Image

In this section, we disclose the warp function that relates the template and the input image. Before that, we introduce the notations and assumptions we made.

2.1 Notations and Assumptions

The deformable surface is explicitly parameterized as a triangulated mesh, as in Fig.1, with N_v vertices $V_i = (x_i, y_i, z_i)^T$, $1 \leq i \leq N_v$. The unknown to be estimated is X , a column vector obtained by concatenating x-,y-,z- coordinates of the N_v vertices. Specifically, $X = [x_1 \ y_1 \ z_1 \ \cdots \ x_{N_v} \ y_{N_v} \ z_{N_v}]^T$. The known 3D mesh corresponding to the template image is denoted by \tilde{X} , thereafter named as template 3D mesh for short. The mesh is composed of N_p patches and N_e edges. For patch j , $1 \leq j \leq N_p$, its three vertices are noted by V_{j1} , V_{j2} and V_{j3} , whose corresponding vertices in the template 3D mesh are \tilde{V}_{j1} , \tilde{V}_{j2} and \tilde{V}_{j3} , respectively. For simplicity, all these vertices are in the camera referential without loss of generality.

The mesh is assumed to be flexible but inextensible, thus preventing the distance between two neighboring vertices from expanding or shrinking too much. We also assume a pinhole perspective camera model, whose intrinsic parameter matrix K is known and kept constant.

2.2 2D-2D Warp Function

The 3D warp of a non-rigid mesh is often modeled as piecewise affine transformation. Specifically, it is usually assumed, as in [3,4,8,6,7], that the barycentric coordinates of each point on its corresponding 3D patch are constant across deformation. The constancy of barycentric coordinates indicates that the surface patches are always planar across deformation, thus one can easily expect that the 2D warp between the template and the input image is piecewise homography. In the following, we present the explicit formulation of the 2D warp.

For any point \tilde{Q} on the j th patch of the template 3D mesh, its coordinates can be expressed as

$$\tilde{Q} = [\tilde{V}_{j1} \ \tilde{V}_{j2} \ \tilde{V}_{j3}] \varepsilon, \quad (1)$$

where ε is the barycentric coordinates of \tilde{Q} on this patch. Its homogeneous projection \tilde{q} on the template image T is:

$$\tilde{\lambda}\tilde{q} = K [\tilde{V}_{j1} \ \tilde{V}_{j2} \ \tilde{V}_{j3}] \varepsilon, \quad (2)$$

with unknown scalar $\tilde{\lambda}$ accounting for the depth.

On the unknown 3D mesh corresponding to the input image I , \tilde{Q} is transferred to Q after some deformation. Since we assume its barycentric coordinates keep constant, the coordinates of Q can be written as

$$Q = [V_{j1} \ V_{j2} \ V_{j3}] \varepsilon, \quad (3)$$

whose homogeneous projection q on the input image I should be:

$$\lambda q = K [V_{j1} \ V_{j2} \ V_{j3}] \varepsilon, \quad (4)$$

where λ is a unknown depth scalar.

Combing eq.(1)-(4), we get

$$(\lambda/\tilde{\lambda})q = K [V_{j1} \ V_{j2} \ V_{j3}] [\tilde{V}_{j1} \ \tilde{V}_{j2} \ \tilde{V}_{j3}]^{-1} K^{-1}\tilde{q}. \quad (5)$$

Considering that V_{jk} and \tilde{V}_{jk} , $1 \leq k \leq 3$ are the vertices of the j th triangular patch, the 3x3 matrix P is invertible, where

$$P = K [V_{j1} \ V_{j2} \ V_{j3}] [\tilde{V}_{j1} \ \tilde{V}_{j2} \ \tilde{V}_{j3}]^{-1} K^{-1}, \quad (6)$$

meaning that the 2D-2D warp P is a homography. Therefore, the template image T and the input image I are correlated by piece-wise homography.

3 Appearance-Based Non-rigid 3D Recovery

Based on the 2D warp, we show how to directly use appearance-based image alignment to recover sparsely textured surface from a single image.

3.1 Direct Lucas-Kanade Method for Non-rigid 3D Recovery

The direct Lucas-Kanade image alignment method [18] is an iterative method to minimize the Sum of Squared Difference (SSD) between the template image T and the input image I by additively adjusting the parameters from a given starting point.

Minimizing SSD. The direct Lucas-Kanade method uses an additive update for the unknown parameter $X \leftarrow X + \Delta X$, where ΔX is the increment in current iteration. The Warp W for the j th patch is P , which can be updated as:

$$P = K [V_{j1} + \Delta V_{j1} \ V_{j2} + \Delta V_{j2} \ V_{j3} + \Delta V_{j3}] [\tilde{V}_{j1} \ \tilde{V}_{j2} \ \tilde{V}_{j3}]^{-1} K^{-1}. \quad (7)$$

Under the assumption of constant intensity, the increment ΔX can be solved by minimizing the SSD energy term $E_{SSD}(X)$:

$$\min_{\Delta X} E_{SSD}(X + \Delta X) = \sum_{j=1}^{N_p} \sum_{u \in C_j} [I(W(u, X + \Delta X)) - T(u)]^2, \quad (8)$$

where C_j is the set of pixels in the image of the j th patch. After using Gauss-Newton approximation, the increment ΔX can be calculated by

$$\Delta X = H_{SSD}^{-1} \left\{ \sum_{j=1}^{N_p} \sum_{u \in C_j} [\nabla I \frac{\partial W}{\partial \Delta X}]^T [T(u) - I(W(u; X))] \right\}, \quad (9)$$

where the Hessian H_{SSD} for the SSD energy term $E_{SSD}(X)$ should be

$$H_{SSD} = \sum_{j=1}^{N_p} \sum_{u \in C_j} [\nabla I \frac{\partial W}{\partial \Delta X}]^T [\nabla I \frac{\partial W}{\partial \Delta X}]. \quad (10)$$

Ambiguity Analysis. According to eq.(6), there are 9 unknowns in the homography, whereas the homography has only 8 independent parameters. Therefore, even assuming perfect alignment for each pixel in the projection of this patch, there is one scalar ambiguity in the estimated patch coordinates. For the whole triangulated mesh, assuming perfect alignment for each patch and considering the connectivity between mesh patches, ideally there is still a global scalar ambiguity in the reconstruction if we only minimize the SSD energy term E_{SSD} as in eq.(8). Actually, we find in our experiment that the Hessian H_{SSD} is minus 1 rank-deficient with some (about one third) close-to-zero eigen-values, demonstrating that monocular recovery of deformable surfaces is an ill-posed problem.

Our observation is consistent with the ambiguity analysis in [19] on the basis of dense and uniform feature correspondences. This is understandable since the correct image alignment can be regarded as establishing extremely dense correspondences, i.e. one feature correspondence for one pixel. Although the image alignment does not better constrain the reconstruction for a well-textured surface, we can indeed expect that it works better for sparsely textured surfaces, the recovery of which becomes more under-constrained when using sparse correspondences only.

Disambiguating Reconstruction. To obtain an unique and stable reconstruction, we introduce constraints on each edge of the mesh by penalizing it from expanding and shrinking too much. For the k th, $1 \leq k \leq N_e$, edge of the mesh defined by two neighboring vertices V_{k1} and V_{k2} , the constraints can be written as: $\|V_{k1} - V_{k2}\| = l_k$, where $\|\cdot\|$ represents L_2 norm, and l_k is the length of k th edge in the template 3D mesh. It can be rearranged into matrix form $\|S_k X\| = l_k$. Rather than using them as hard constraints, we minimize the equivalent side-length energy term $E_s(X)$, which is defined by

$$E_s(X) = \sum_{k=1}^{N_e} (\|S_k X\|^2 - l_k^2)^2. \quad (11)$$

Combined with the SSD energy term $E_{SSD}(X)$, the direct Lucas-Kanade image alignment method can be formulated as:

$$\min_{\Delta X} \{E_{SSD}(X + \Delta X) + \omega_s E_s(X + \Delta X)\}, \quad (12)$$

where ω_s is a user-defined weighting factor. Using Gauss-Newton approximation, the increment ΔX can be easily calculated.

Without introducing any *a priori* knowledge that tend to oversmooth sharp details, our appearance-based method can be used to accurately recover inextensible surfaces with poor texture and sharp creases.

From eq.(10), we can see that the Jacobian and the Hessian should be recomputed for each pixel at each iteration, since they are evaluated at current estimation of the vertex parameters X . Generally it is computationally demanding. In the following, we show how to accelerate this direct Lucas-Kanade method by using ICIA, in which the most time-consuming part can be precomputed.

3.2 Efficient Image Alignment for Non-rigid 3D Recovery

Combining Additive and Inverse Compositional Expressions. In ICIA [17], the warp is updated by $W \leftarrow \bar{W} \circ (\Delta W)^{-1}$, where the operator ‘ \circ ’ means the composition of the current warp \bar{W} and the increment warp ΔW . Specifically, for a homography warp, it can be updated by $P \leftarrow \bar{P}(I + \Delta P)^{-1}$, where $\Delta W = I + \Delta P$ is the incremental homography warp, and $\bar{W} = \bar{P}$ is the current homography.

Since we need to estimate the vertex coordinates embedded in the homography, rather than the homography in itself, we have to devise the update rule for

the mesh parameters X . Same as the direct Lucas-Kanade method, we use an additive update rule for X , i.e. $X \leftarrow X + \Delta X$. To make the warp updated from the inverse composition equivalent to that from the additive updating in eq.(7), we let the following equation hold:

$$P = \bar{P}(I + \Delta P)^{-1}, \quad (13)$$

where P is from eq.(7), while \bar{P} from eq.(6). This rule has been used in [20] for fast surface reconstruction from stereo. Note that it is not completely the same as the original ICIA image alignment in [17], since the parameter X can be directly updated through the additive rule. In the following, we still name our method as an ICIA method, considering that the homography warp is updated by inverse composition.

It is obvious that when $\Delta X \rightarrow 0$, the incremental warp $\Delta W \rightarrow I$, which means that it is an identity warp. Before giving the explicit relationship between ΔP and ΔX , we first show how to use the ICIA method in non-rigid 3D recovery.

According to [17], image alignment can alternatively be formulated as:

$$\min_{\Delta X} E_{SSD}(X + \Delta X) = \sum_{j=1}^{N_p} \sum_{u \in C_j} [T(\Delta W(u, \Delta X)) - I(W(u, X))]^2. \quad (14)$$

Using Gauss-Newton approximation, the increment ΔX can be derived from:

$$\Delta X = H_{SSD}^{-1} \left\{ \sum_{j=1}^{N_p} \sum_{u \in C_j} [\nabla T \frac{\partial \Delta W}{\partial \Delta p} \frac{\partial \Delta p}{\partial \Delta X}]^T [T(u) - I(W(u; X))] \right\}, \quad (15)$$

where Δp represents the elements in ΔP , and the Hessian

$$H_{SSD} = \sum_{j=1}^{N_p} \sum_{u \in C_j} [\nabla T \frac{\partial \Delta W}{\partial \Delta p} \frac{\partial \Delta p}{\partial \Delta X}]^T [\nabla T \frac{\partial \Delta W}{\partial \Delta p} \frac{\partial \Delta p}{\partial \Delta X}]. \quad (16)$$

To calculate H_{SSD} , we need to compute $\partial \Delta p / \Delta X$, which is presented in the following subsection.

Computing $\partial \Delta p / \Delta X$. When $\Delta P \rightarrow 0$, the inverse of the incremental warp can be approximated (first order approximation) by

$$(I + \Delta P)^{-1} = I - \Delta P. \quad (17)$$

From eq.(13) and eq.(17), we can calculate ΔP as follows:

$$\Delta P = -K [\tilde{V}_{j1} \ \tilde{V}_{j2} \ \tilde{V}_{j3}] [V_{j1} \ V_{j2} \ V_{j3}]^{-1} [\Delta V_{j1} \ \Delta V_{j2} \ \Delta V_{j3}] [\tilde{V}_{j1} \ \tilde{V}_{j2} \ \tilde{V}_{j3}]^{-1} K^{-1}, \quad (18)$$

from which, the $\partial \Delta p / \Delta X$ can be straightforwardly computed, since Δp is a linear function of ΔX .

Efficiency Analysis. From eq.(16), the gradient of the template image ∇T and that of the increment warp $\partial\Delta W/\Delta p$, i.e. the most time-consuming pixel-related parts of the Hessian, are constant across iteration, since they are evaluated at $\Delta X = 0$. However, $\partial\Delta p/\Delta X$ is dependent on the current estimation of X , thus should be recomputed at each iteration. Fortunately, it is irrelevant to pixel coordinates, and needs only to be recomputed for each patch. Specifically,

$$H_{SSD} = \sum_{j=1}^{N_p} \left(\frac{\partial\Delta p}{\partial\Delta X} \right)^T H_{const} \frac{\partial\Delta p}{\partial\Delta X}, \quad (19)$$

where H_{const} is the constant part of the Hessian,

$$H_{const} = \sum_{u \in C_j} \left[\nabla T \frac{\partial\Delta W}{\partial\Delta p} \right]^T \left[\nabla T \frac{\partial\Delta W}{\partial\Delta p} \right]. \quad (20)$$

To disambiguate the reconstruction, we should introduce the side-length energy term $E_s(X)$ as in section 3.1. The Hessian for this term should also be recomputed. However, the number of sides is always much smaller than that of pixels, thus can be evaluated very fast.

4 Fusing Features and Appearance

The appearance-based image alignment methods are usually sensitive to disturbance on the pixel intensity. When lighting changes or small occlusion occurs, it is helpful to fuse feature correspondences and appearance-based image alignment [14], since these feature points, serving somewhat as anchors, are able to prevent the mesh from drifting. In addition, introducing feature correspondences poses little increase in computational burden, since the Hessian for this part can be easily computed. The feature set used here is the inlier set from the feature-based methods whose reprojection error is lower than 1 pixel. The feature energy term $E_f(X)$ is measured by

$$E_f(X) = \|MX\|^2, \quad (21)$$

where M is the structure matrix constructed by following [6,7,8]. Specifically, we simultaneously minimize the SSD energy term $E_{SSD}(X)$, the side-length term $E_s(X)$ and the feature energy term $E_f(X)$:

$$\min_{\Delta X} \{ E_{SSD}(X + \Delta X) + \omega_s E_s(X + \Delta X) + \omega_f E_f(X + \Delta X) \}, \quad (22)$$

where ω_f is a user-defined weighting factor. This equation can be easily solved by using Gauss-Newton approximation. Note that the Hessian for the SSD energy term E_{SSD} can be calculated either by eq.(10) in the direct Lucas-Kanade method or by eq.(19) in the ICIA method.

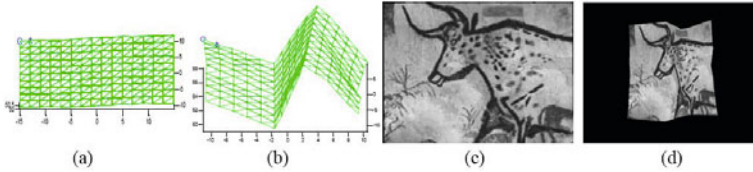


Fig. 2. Synthetic 100-frame mesh sequence. (a) The mesh in rest position. (b) The 50th frame with largest deformation. The sequence is retextured by using sparse texture (c), and reprojected onto images (d) by a virtual camera.

5 Experimental Results

In this section, we use both synthetic data and real images to test the performance of our proposed methods. For feature-based methods, we use 60 modes for the global deformation model [6] and 20 modes for each of the local deformation model [8].

5.1 Synthetic Data

We generate a 100-frame sequence of a piece of paper with sharp creased deformation by using motion capture devices (Fig.2(a,b)), then we synthesize sparse texture (Fig.2(c)) on the meshes, which are backprojected onto 2D images using a synthetic projective camera (Fig.2(d)). The mesh resolution is 11×16 , and size is 200mm x 300mm.

Convergence w.r.t. Rough Initialization and Intensity Noise. Here, we use the 50th frame as the target, whose deformation is the largest in the sequence. We add zero mean Gaussian noise with deviation σ on the ground-truth 3D mesh to simulate rough initialization. Both the template and the input image are corrupted by zero mean Gaussian noise with deviation 2 grey levels. 100 sparse feature correspondences are also randomly generated for the fusion methods. We measure the average vertex-to-vertex error between the ground-truth 3D mesh and the estimated 3D mesh from image alignment after 20 iterations. The result is said to be convergent when the average 3D error is lower than 2mm. We compare the performance of the direct Lucas-Kanade method (DLK), the Inverse Compositional Image Alignment method (ICIA), and their fusion with features, shown in Fig.3(a) as (F+DLK) and (F+ICIA), respectively. We vary σ from 0.4 mm to 4 mm, and repeat each method for 500 times at each noise level. From Fig.3(a), we can see that the DLK and the ICIA have almost the same performance, which is understandable since they are almost equivalent. When noise is large, the ICIA method is slightly weaker than the DLK method due to the first-order approximation used in eq.(17). Both methods can be improved by fusing feature correspondences.

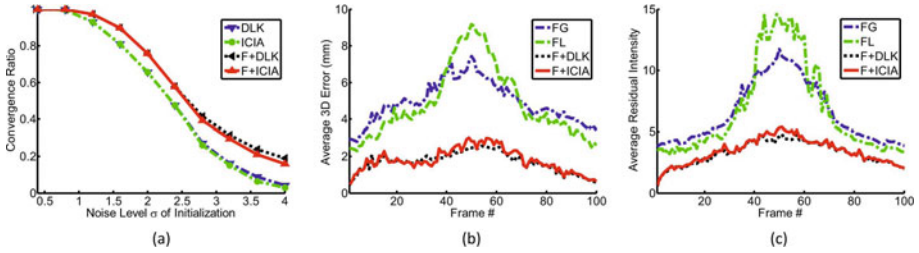


Fig. 3. Synthetic experiments. (a) Use the 50th frame (with largest deformation) to test the convergence performance w.r.t. rough initialization with intensity noise. (b) The average 3D vertex-to-vertex error for the whole 100-frame sequence from four different methods. (c) The average intensity of the residual image for four different methods. We can see that when large deformation occurs (in the middle of the sequence), the feature-based methods (FG) and (FL) can not accurately recover the mesh, which can be improved by our fusion methods (F+DLK) and (F+ICIA).

Table 1. Time Performance in direct Lucas-Kanade (DLK) and ICIA methods

Methods	Precomputation	Compute Hessian	One Iteration
DLK	-	2.827s	3.462s
ICIA	2.672s	0.212s	0.718s

In the following, we initialize the appearance-based methods by using the rough results from feature-based methods. Considering that the fusion methods work better, we shall only use the two fusion ones (F+DLK) and (F+ICIA).

Improving Feature-Based Methods by Fusion. Here we use SIFT [21] to establish 3D-2D feature correspondences for the whole sequence, and follow the feature-based closed-form solution with global deformation model (FG) [6] and the convex method with local deformation model (FL) [8] to get rough initialization. The fusion methods (F+DLK) and (F+ICIA) are initialized by both feature-based methods (FG) and (FL), and only the results with less residual intensity are presented. Both the template and the input image are corrupted by Gaussian noise with zero mean and deviation 2 gray levels. Fig.3(b) shows the average 3D vertex-to-vertex error and Fig.3(c) the average intensity on the residual image. The results of the 50th frame are also presented in Fig.4. From these results, we can see that, compared with the local deformation model, the global deformation model is more likely to oversmooth the sharp creases. However, when large deformation occurs, the shape from the local deformation model is somewhat irregular. This is due to the fact that the local deformation model relies not so heavily on the training data. These problems can be reliably remedied by our fusion methods, both of which have almost the same performance.

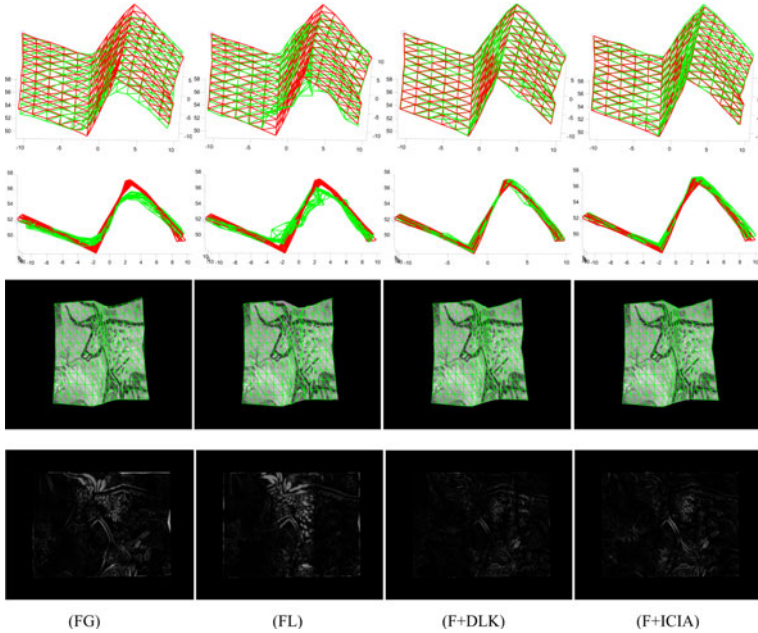


Fig. 4. Visual comparison of the results from different methods for the 50th frame with largest deformation. From left to right are the results from the global deformation model (FG), the local deformation model (FL), the fusion method with direct Lucas-Kanade (F+DLK), and the fusion one with ICIA (F+ICIA), respectively. From top to bottom are the recovered 3D mesh(green) together with the ground truth(red), the same 3D meshes observed from side view, the input image overlaid by the mesh projection, and the residual images, respectively.

Time Performance. We implement all the methods in a 1.6GHz laptop with 2GB RAM by using MATLAB. The image resolution is 720x576, and the mesh size is 11x16 with 528 variables, 475 edges and 300 patches. The time performance is shown in Table 1 for one iteration averaged from 50 iterations. The precomputation of the constant part of the Hessian for the ICIA method takes 2.672s. It only takes 0.212s in the ICIA method to compute the full Hessian, in contrast to 2.827s in the direct Lucas-Kanade method, which is about 14 times faster. The ICIA method takes 0.718s in one iteration, while the direct Lucas-Kanade method takes 3.462s, with an acceleration rate about 5 times. The acceleration rate shrinks, because some other pixel-related process, like bilinear interpolation, takes about 0.4s, which is a bottleneck when using MATLAB.

5.2 Real Images

We also have some preliminary results on a piece of paper with sparse texture and sharply creased deformation. To show that our methods can accurately

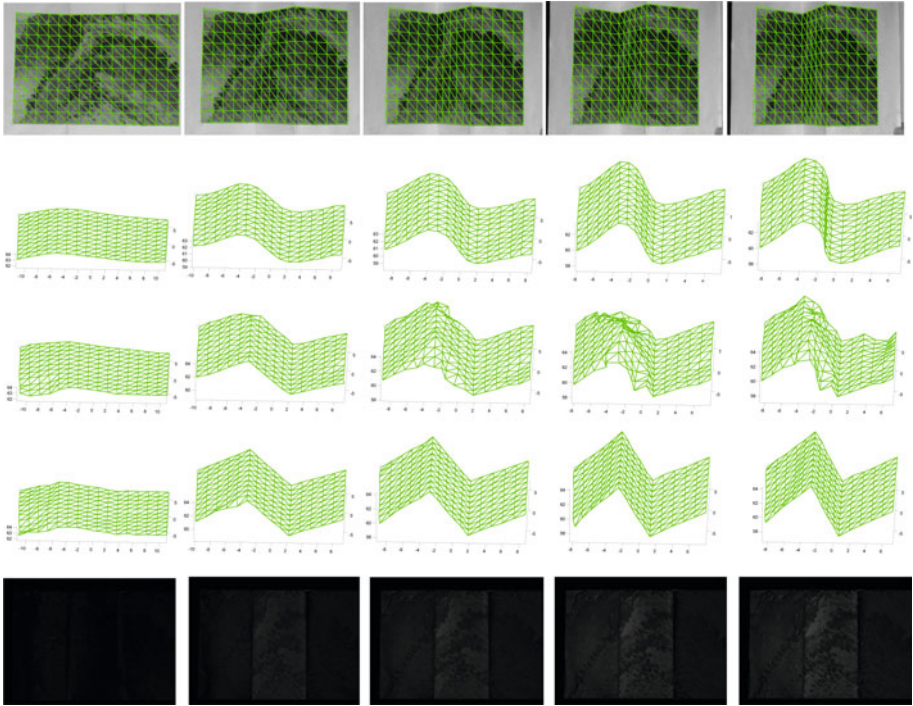


Fig. 5. Five frames of a piece of paper with sparse texture and sharp creases. First row: Images overlaid by the projection of the mesh reconstructed from our method. From 2nd to 4th row: 3D meshes from the Global deformation model (FG), Local deformation model (FL), and our fusion method (F+ICIA), respectively. Last row: The residual images after image alignment.

recover the sharp creases, we intentionally make the creases coincident with the mesh edges, otherwise the recovered creases would be smoothed due to surface discretization. The images are captured by a FLea2 camera with 800x600 resolution. Generally the matched 3D-2D feature pairs are less than 200. Considering the efficiency of the (F+ICIA) method and its equivalence to the (F+DLK) method, we only use the (F+ICIA) method here, which is initialized by the (FL) method. From Fig.5, we can see again that the global deformation model can only approximate the sharply creased surface, while the shape from the local deformation model is somewhat irregular due to severe lack of features, although not so seriously being oversmoothed. By observing the residual images in the last row, we see that our fusion method works well in case of local intensity changes caused by significant variation in surface orientation. We also show our method (F+ICIA) can improve the photo-consistency of the results from the (FL) method [\[8\]](#) for a piece of cloth with relatively dense texture and sharp folds, which can be concluded by comparing the residual images in Fig.6.

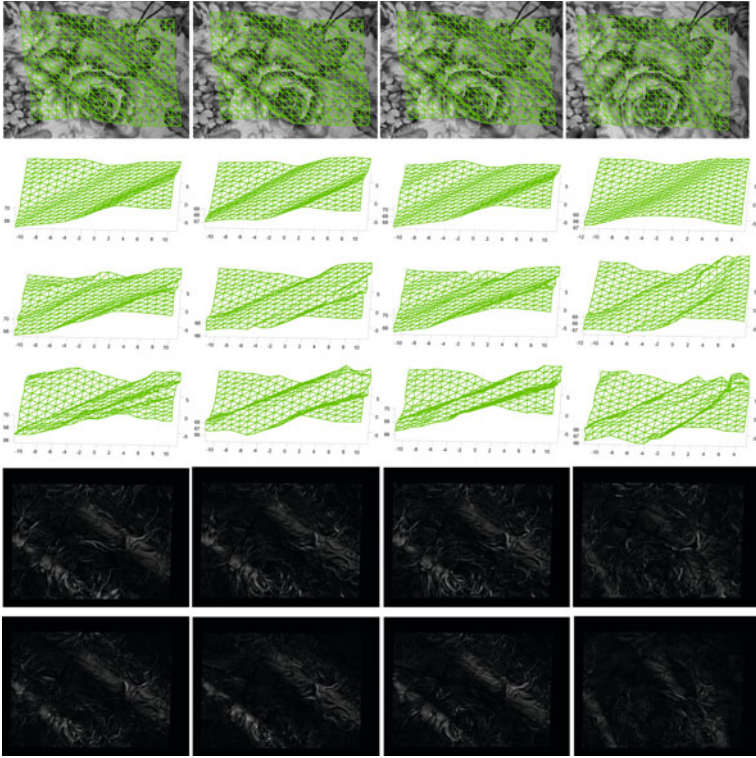


Fig. 6. Four frames of a piece of cloth with relatively dense texture and sharply folded deformation. First row: Images overlaid by the projection of the mesh reconstructed from our method (F+ICIA). From 2nd to 4th row: 3D meshes from Global deformation model (FG), Local deformation model (FL), and our method (F+ICIA), respectively. From 5th to 6th row: The residual images for (FL) and for (F+ICIA), respectively.

6 Conclusion

We have shown how to efficiently refine the 3D structure of poorly textured nonrigid surfaces, even with sharp details, from a single image by fusing feature correspondences and appearance-based image alignment. To our knowledge, this work is the first one that can accurately recover sharply creased surfaces in case of sparse texture.

We should mention that it is still a challenging task to recover sharp details in case of large occlusion. When large occlusion occurs, it is inevitable to introduce some prior knowledge, which tends to oversmooth sharp details. In addition, we partially conquer the disturbance on pixel intensity by fusing features, which is insufficient in case of large lighting changes. The potential lighting variation can be further compensated, by using the Dual ICIA method [22] for efficiency, which is left to the future.

Acknowledgement. This work was partly supported by Grant-in-Aid for Scientific Research (21240015) from the Japan Society for the Promotion of Science.

References

1. Cohen, L., Cohen, I.: Finite element methods for active contour models and balloons for 2d and 3d images. *PAMI* 15, 1131–1147 (1993)
2. Bhat, K.S., Twigg, C.D., Hodgins, J.K., Khosla, P.K., Popovic, Z., Seitz, S.M.: Estimating cloth simulation parameters from video. In: *ACM Symposium on Computer Animation* (2003)
3. Salzmann, M., Hartley, R., Fua, P.: Convex optimization for deformable surface 3d tracking. In: *ICCV* (2007)
4. Zhu, J., Hoi, S.C.H., Xu, Z., Lyu, M.R.: An effective approach to 3d deformable surface tracking. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part III*. LNCS, vol. 5304, pp. 766–779. Springer, Heidelberg (2008)
5. Varol, A., Salzmann, M., Tola, E., Fua, P.: Template-free monocular reconstruction of deformable surfaces. In: *ICCV* (2009)
6. Salzmann, M., Moreno-Noguer, F., Lepetit, V., Fua, P.: Closed-form solution to non-rigid 3d surface registration. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part IV*. LNCS, vol. 5305, pp. 581–594. Springer, Heidelberg (2008)
7. Moreno-Noguer, F., Salzmann, M., Lepetit, V., Fua, P.: Capturing 3d stretchable surfaces from single images in closed form. In: *CVPR* (2009)
8. Salzmann, M., Fua, P.: Reconstructing sharply folding surfaces: a convex formulation. In: *CVPR* (2009)
9. Ilic, S., Salzmann, M., Fua, P.: Implicit meshes for effective silhouette handling. *IJCV* 72, 159–178 (2007)
10. Terzopoulos, D., Metaxas, D.: Dynamic 3d models with local and global deformations: deformable superquadrics. *PAMI* 13, 703–714 (1991)
11. Vidal, R., Hartley, R.: Perspective nonrigid shape and motion recovery. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part I*. LNCS, vol. 5302, pp. 276–289. Springer, Heidelberg (2008)
12. Xiao, J., Kanade, T.: Uncalibrated perspective reconstruction of deformable structures. In: *ICCV* (2005)
13. Ecker, A., Jepson, A., Kutulakos, K.: Semidefinite programming heuristics for surface reconstruction ambiguities. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part I*. LNCS, vol. 5302, pp. 127–140. Springer, Heidelberg (2008)
14. Zhu, J., Lyu, M.R., Huang, T.S.: A fast 2d shape recovery approach by fusing features and appearance. *PAMI* 31, 1210–1224 (2009)
15. Pilet, J., Lepetit, V., Fua, P.: Fast non-rigid surface detection, registration and realistic augmentation. *IJCV* 76, 109–122 (2008)
16. Munoz, E., Buenaposada, J., Baumela, L.: Efficient model-based 3d tracking of deformable objects. In: *ICCV* (2005)
17. Baker, S., Matthews, I.: Lucas-kanade 20 years on: A unifying framework. *IJCV* 56, 221–255 (2004)
18. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. *IJCAI* (1981)
19. Salzmann, M., Lepetit, V., Fua, P.: Deformable surface tracking ambiguities. In: *CVPR* (2007)
20. Sugimoto, S., Okutomi, M.: A direct and efficient method for piecewise-planar surface reconstruction. In: *CVPR* (2007)
21. Lowe, D.: Distinctive image features from scale-invariant keypoints. *IJCV* 20, 91–110 (2004)
22. Bartoli, A.: Groupwise geometric and photometric direct image registration. *PAMI* 30, 2098–2108 (2008)

Local Empirical Templates and Density Ratios for People Counting

Dao Huu Hung¹, Sheng-Luen Chung¹, and Gee-Sern Hsu²

¹ Department of Electrical Engineering

² Department of Mechanical Engineering

National Taiwan University of Science and Technology

Abstract. We extract local empirical templates and density ratios from a large collection of surveillance videos, and develop a fast and low-cost scheme for people counting. The local empirical templates are extracted by clustering the foregrounds induced by single pedestrians with similar features in silhouettes. The density ratio is obtained by comparing the size of the foreground induced by a group of pedestrians to that of the local empirical template considered the most appropriate for the region where the group foreground is captured. Because of the local scale normalization between sizes, the density ratio appears to have a bound closely related to the number of pedestrians that induce the group foreground. We estimate the bounds of density ratios for groups of different numbers of pedestrians in the learning phase, and use the estimated bounds to count the pedestrians in online settings. The results are promising.

1 Introduction

People counting is one the central issues considered in the field of intelligent video surveillance (IVS), although its application scope goes beyond surveillance. A few approaches were proposed in the last decade, and will be reviewed later in this paper. Because of expensive computation, some of the existing methods cannot perform in real time. We propose a method able to perform in real time. It consists of two phases: offline learning and online counting. In the offline learning phase, we extract the templates of single pedestrians from a large collection of video samples taken under different viewpoints, distances, depths of views, and movement patterns. Although these empirical templates vary in size as the pedestrians move across the scene, in different local regions the size ratios between the foregrounds made by group pedestrians and the empirical templates appear to be bounded. We call these bounds *density ratio bounds*, and the single-pedestrian templates *local empirical templates*. From our experiments the density ratio bounds seem to be robust to viewpoint changes and size variations across the scene. In the online counting phase, both of the local empirical templates and density ratio bounds are used to estimate the number of people in a foreground extracted from a scene.

Two assumptions are needed for the proposed method to work: (1) pedestrians must be in upright pose; (2) no vehicles or other moving objects appear in the scene, which means that people are the only moving objects considered. A few cases are also excluded in our study: (1) pedestrians far away from the camera so their sizes appear very small¹; (2) groups with serious occlusion which even challenge human eyes to count.

The proposed methods are evaluated on both PETS 2009 benchmark datasets and our in-house video samples collected from real scenes with various parameters, such as viewpoints and occlusions.

The rest of this paper is organized as follows. Section 2 reviews related works. The proposed method is presented in detail in Section 3. The performance evaluation and comparison with other methods are given in Section 4, followed by the conclusion and possible future research in Section 5.

2 Related Works

To count people in entrance/exit gates and in elevator zones, overhead view was usually used [1,2,3]. There is no occluded people in this viewpoint thus it is easy to segment and count individuals. However, the region of interest is limited by the constraint of ceilings. Through the training, Park et al. [4] obtained the mean and variance values of person's size in each sector of 72-sector-divided images which were sensitive to camera height and were used to count people later. His method only work for overhead viewpoint cases.

Haritaoglu et al. [5] developed W4 system for real-time detecting and tracking multiple people. It counts groups of people by roughly finding heads through corresponding peaks of vertical projected histogram. Human shape models were used to interpret the foreground in a Bayesian framework that was implemented by Markov Chain Monte Carlo method [6]. It could segment a group into individuals at the expense of high computational cost.

Human appearance models were used to detect people in occlusion. Elgammal and Davis [7] built appearance models for unoccluded people entering the scene and subsequently tracked them in occlusion. Xi Zhao *et al.* [8] presented a people counting approach based on face detection and tracking. A standard face detector located faces and tracked them. Free camera viewpoint is achieved but people need to turn their faces to cameras. Li *et al.* [9] trained offline Adaboost HOG (Histogram of Oriented Gradients) features of heads and shoulders to detect people in each frame.

Multiple-camera and stereo solution is another class of approaches to resolve the problem of occlusion. Kelly et al. [10] discussed a stereo solution for counting people in both indoor and outdoor crowded scenes under various viewpoints. They developed 3D clustering process by using bio-metrically inspired constraints for people detection and track matching process by using a weighted maximum cardinality matching scheme. However, in general multiple cameras

¹ Quantitatively, the height of the pedestrian is less than 5% height of the view.

and stereo solutions require prior deliberate calibration and a significant amount of work for registration.

For dense groups in crowded areas, foreground may not be easily segmented. Davies et al. [11] used linear fitting to find the relationship between the number of edge pixels and the number of moving people in a region of interest. Texture was measured as different qualitative labels since they argued that images of sparse and dense crowds were often made up of low and high frequency patterns, respectively [12]. The link between these qualitative labels and the count depends on specific applications. The classification was done by self-organizing map that involved an intensive training. Kilambi *et al.* [13] provided a solution in the light of using geometric projections, dealing with the entire area occupied by a group as a whole rather than trying to detect individuals separately. Estimated occupied areas were combined with some social statistics of interpersonal distance to determine the count. Chan et al. [14] adopted Gaussian process regression for segment, internal edge and texture features which are normalized to account for perspective to estimate the number of people. Albiol *et al.* [15] analyzed moving corners to count people. They assumed that each person, on average, has a particular number of moving corners. However, it does not hold in general since the average number of moving corners per person may vary in accordance of tilt angle and distance from people to cameras.

In this paper, we take advantage of treating a group of people as a whole. Our system consists of low cost but effective enough modules in order to ensure both the real-time performance and good accuracy.

3 Proposed Method

The proposed method is composed of an offline learning phase and an online counting phase. In the offline learning phase, we extract foregrounds from a large collection of videos taken from surveillance cameras with various viewpoints and viewing areas, and under different weather conditions. Clustering on the extracted foregrounds leads to the generation of *local empirical templates* and *density ratio bounds*. The former are templates for the foregrounds induced by single pedestrians, and the latter can be used to estimate the number of pedestrians in the foregrounds induced by multiple pedestrians. In the online counting phase, the LET and the density ratio bounds are both used to estimate the number of pedestrians in a foreground captured online.

3.1 Offline Learning and Local Empirical Template Extraction

A large set of videos is collected from stationary surveillance cameras installed at different locations, with different viewpoints and various fields of views, and under different weather conditions. The Gaussian Mixture Model (GMM) proposed by Stauffer and Grimson [16] is applied to extract foregrounds from these videos. The extracted foregrounds are inspected manually, and corrected in cases when the foregrounds fail to be accurately detected by the GMM approach. We split

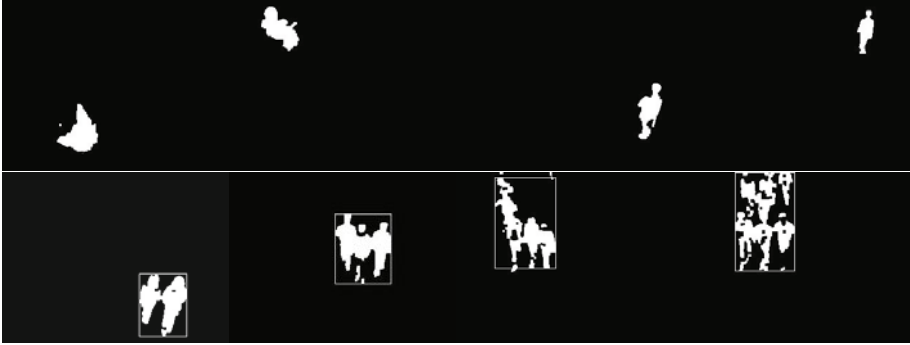


Fig. 1. Examples of single foregrounds (first row) and group foregrounds (the last row)

the foregrounds into two categories: *single foregrounds* and *group foregrounds*. The former are contributed by single pedestrians, and the latter are induced by two or more of pedestrians that generate overlapping foregrounds. The overlapping foregrounds, or group foregrounds, can be caused by pedestrians with close proximity to each other or pedestrians parted away from each other but with occlusion from the view of the camera. Examples of single foregrounds and group foregrounds are shown in Fig. 1.

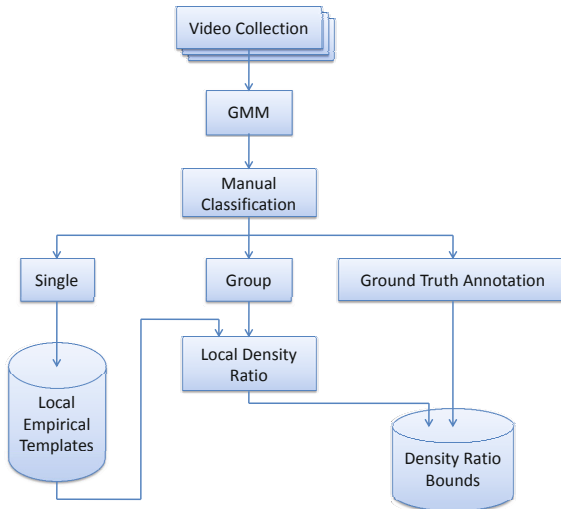


Fig. 2. Flowchart of the offline learning phase

The *local empirical templates* of single foregrounds are represented by their width, height, and trajectories or their positions in the image. Depending on different settings, especially the viewpoints of the camera, the number of LET in a fixed-view window can be as few as a couple or as many as tens. Experiments on the single LET and group foregrounds reveal the following observations:

- The LET of single foregrounds can be used to discriminate single foregrounds from group foregrounds. The relative sizes of the extracted foregrounds from each other reveal the corresponding crowd densities in many cases, and therefore the foregrounds with sizes smaller than most of the others are likely to be caused by single pedestrians. The decision can be made using a distance measure between the foreground and the LET.

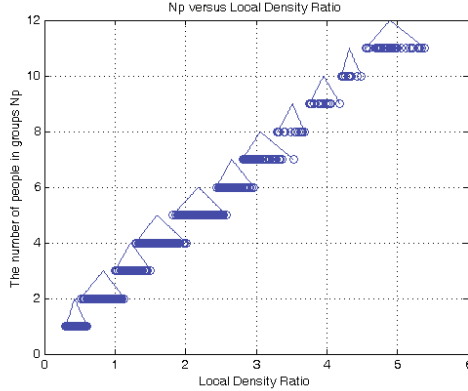


Fig. 3. Local density ratio versus the number of people N_p

- If the viewing window is divided into $M \times N$ cells by a grid, as shown in Fig. 3c, the *local density ratio*, $D(m, n)$, can be defined for each cell (m, n) , $n = 1, \dots, N; m = 1, \dots, M$, as follows,

$$D(m, n) = \frac{S_g(m, n)}{\mathbf{T}_s(m, n)} = \frac{S_g(m, n)}{H_{temp}(m, n) \times W_{temp}(m, n)} \quad (1)$$

where $S_g(m, n)$ is the size of a group foreground captured at cell (m, n) and its neighbors because a group foreground may not appear in one cell, and $\mathbf{T}_s(m, n)$ is the size of the local empirical template, measured by its width $W_{temp}(m, n)$ and height $H_{temp}(m, n)$ at the cell (m, n) . It is observed that although both $S_g(m, n)$ and $\mathbf{T}_s(m, n)$ vary across the viewing window, the variation in $D(m, n)$ appears limited by a bounded range when the crowd density in the group foreground is kept a constant. In other words, the following bounds can be observed,

$$D_M(N_p) > D(m, n, N_p) > D_m(N_p) \quad (2)$$

where D_M and D_m are the upper and lower bounds of density ratio $D(m, n, N_p)$ of a group foreground containing N_p pedestrians at cell (m, n) .

Eq. 2 shows that the local density ratios seem independent of the cell’s location (m, n) , and depend on N_p only. Because N_p can be considered an absolute crowd density of a group foreground which has different sizes over the viewing window,

Table 1. Bounds of local density ratio and corresponding count of a group of people

Local Density Bounds	N_p
0.3 ~ 0.6	1
0.6 ~ 1.1	2
1.1 ~ 1.4	3
1.4 ~ 1.9	4
1.9 ~ 2.45	5
2.45 ~ 2.85	6
2.85 ~ 3.3	7
3.3 ~ 3.7	8
3.7 ~ 4.2	9
4.2 ~ 4.7	10
> 4.7	11

the local density ratio $D(m, n, N_p)$ normalizes its size variation to that of the LET.

The overall offline learning phase can be summarized by the flowchart shown in Fig. 2.

From our collection of videos, local density ratios of many group foregrounds which are different from the number of people, N_p , in groups are computed. The relationship between the number of people in groups and their local density ratios are obtained and sketched in Fig. 3. Given a fixed number of N_p , we consider the distribution of local density ratios as a triangle. $[D_M, D_m]$, the bounds of local density ratios for various N_p are found at the intersections of these triangles and given in Table 1.

3.2 Online Counting

The online counting phase is composed of the following steps:

1. Foregrounds are firstly extracted from the input video using the GMM [16], similar to the first step in the offline learning.
2. A nearest neighbor classifier trained in the offline learning phase by using a feature vector formed by the height-normalized size and the normal vectors extracted along the smoothed contour of a foreground is used to classify single foregrounds from group foregrounds in the online counting phase. This classification is based on multiple instances captured across a few successive frames. LET of test videos are manually initialized like in the learning phase if the knowledge of test videos is available, or are self-discovered by the following scene-based template update mechanism.
3. The single foregrounds, their trajectories, and the LET that have validated the single foregrounds are kept in a memory buffer for the cells where the single foregrounds are captured. That is, these single foregrounds are used to update corresponding LET, so-called scene-based template update, according to the following formulae.

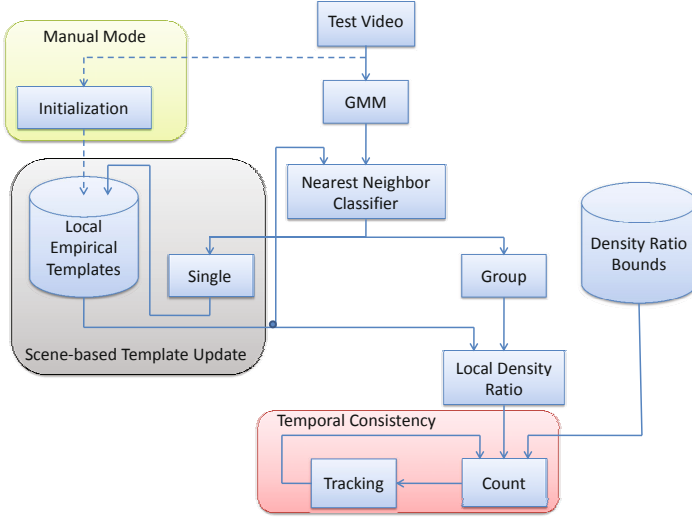


Fig. 4. Flowchart of the online counting phase

$$H_{temp}(new)(m, n) = (1 - \alpha)H_{temp}(old)(m, n) + \alpha H_{sing}(m, n) \quad (3)$$

$$W_{temp}(new)(m, n) = (1 - \alpha)W_{temp}(old)(m, n) + \alpha W_{sing}(m, n) \quad (4)$$

where, $H_{sing}(m,n)$ & $W_{sing}(m,n)$ are height and width of a detected single foreground, respectively, $H_{temp}(m,n)$ & $W_{temp}(m,n)$ are sizes of LET at the cell (m,n) , and α is the learning rate. Because single foregrounds may not appear all over the viewing window, interpolations and extrapolations on sections of their trajectories are performed to estimate and extend the most part of regions that foregrounds appear. It is not a rare condition that single foregrounds only appear in certain segments of a walkway because of occlusion, merging, and low contrast to the backgrounds, etc. Therefore, some cells are short of LET, and some LET's trajectories can be broken or segmented. In the online counting phase, trajectories of both single and group foregrounds will be kept in the buffer and analyzed to map out walkway regions. When single foregrounds appear in segments of these regions, interpolation and/or extrapolation based on the observed single foregrounds will be performed to fill in the cells with "virtual" single foregrounds passing through. This step helps to distinguish the regions with foregrounds from the rest without foregrounds, and establish the scene-based spatial distribution of LET with appropriate sizes.

4. With the established scene-based spatial distribution of the LET, the count of pedestrians in a foreground captured in a local cell (m, n) on the viewing window can be estimated by the local density ratio in Eq. (3) with Table 1. Because local density ratio is computed per frame at each cell, each cell will end up with one to a few local density ratios when a foreground moves

through. The majority of these density ratios are averaged and considered as the density ratio of the cell. Together with the density ratios evaluated at all cells where the foreground passes, the density ratio of the foreground can be properly determined by a majority voting. To ensure the accuracy of the people count on the foreground, the current count is checked for consistency with the counts obtained along the trajectories of the foregrounds appeared in the previous frames. Possible split and merge of foregrounds are also considered in this consistency check.

The above online counting can be summarized in the flowchart in Fig. 4.

4 Experimental Results and Performance Evaluation

The proposed method is evaluated using both of the PETS2009 benchmark data and our own collection of test data. The implementation is on the Visual C++ platform with libraies from OpenCV 1.1, running upon an Intel core 2 Duo T9300 2.5 GHz.

4.1 Evaluation on PETS2009 Benchmark

PETS2009 benchmark provides a training dataset S0, containing subsets for background model learning. Frames in dataset S0 contain people walking through the scene. Therefore, we exploit sizes of these people in dataset S0 for initializing local empirical templates which are used in the online counting phase.

Fig. 5 shows some typical visual results of testing on both subsets in view_001 of dataset S1, L1. The number at the bottom of each bounding box is its estimated count. The number in the top-left corner of the image is the total estimated count throughout entire image. Manually counted ground truth of each frame is compared with the total estimated counts of our proposed method, of moving corners-based method [15], and of holistic properties-based method [14], tested on the same dataset. These results are adapted from their papers [14,15]. Fig. 6 depicts the comparison by graphs sketched in the same coordinate.

Results tested on subset Time_13-59 seem to be better than those tested on subset Time_13-57, since it contains fewer patterns of fully and nearly fully occluded people. In comparison with other methods tested on the same benchmark [14,15,17,18], the results of our proposed method are competitive.

4.2 Evaluation on In-House Collection

In this section, we further assess the performance of the proposed method against various viewpoints, brightness illumination, and arbitrary human movement, etc. via six video samples. Fig. 7 shows some sample frames and the counting results.

The first row of Fig. 7 shows a scene recorded at noon. This scene is challenging since the left hand side of the scene is under strong sunshine and the right one is much darker. Its background contains a lot of texture. When a pedestrian

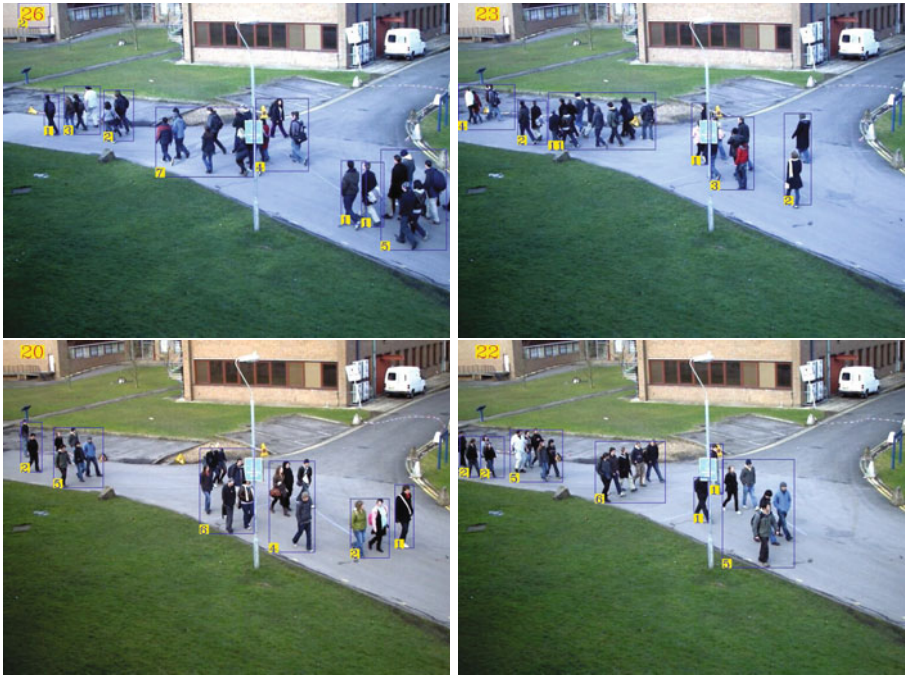


Fig. 5. Results of the proposed approach on dataset S1, L1 (view_001); the first row is of subset Time_13-57, the second row is of subset Time_13-59

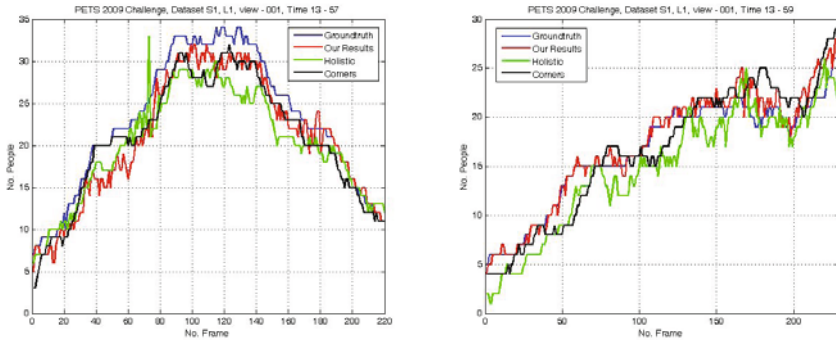


Fig. 6. Comparison between the results of our proposed approach, of methods of using moving corners [15], and of method of using Holistic properties [14], tested on the same datasets, and ground truth. These results are adapted from their papers [14,15]. The left graphs are for subset Time_13-57, the right ones are for subset Time_13-59.

move from bright to dark regions and vice versa, the foreground pattern changes considerably that causes difficulty. However, our approach can handle this situation since local empirical templates in these regions are different. Therefore, shadow seems to be tolerated in the left hand side of the image, even when people are in occlusion. In our system, we also incorporate an algorithm to remove moving cast shadow [19]. The first row shows two pairs of resulting images in two cases of using and not using the algorithm of moving cast shadow elimination.



Fig. 7. Typical visual results of our proposed approach tested on our video collection

The next two rows of Fig. 7 show a scenario under different tilt angles, including the overhead viewpoint. The fifth row illustrates some groups of people are far from camera. The last row shows a crowded scene of a road intersection, containing large groups of people in occlusion. In the road intersection, pedestrians really move in unconstrained fashions. Although good results are often observed, some false positive and false negative occur in the situations of nearly full and full occlusion and of moving bicycles.

5 Concluding Remarks and Future Works

We have presented a method that uses knowledge of single foregrounds to estimate the number of people in group foregrounds. It is the fact that density of group foregrounds varies according to that of single foregrounds in a local region of the image. The ratio of density of group foregrounds which contain a same number of people to that of corresponding single foregrounds falling into a particular bounded range is proved by a large collection of videos covering various viewpoints and scenarios, and illumination conditions. Bounds of the local density ratio obtained in the offline learning phase and local empirical templates are used in the online counting phase. Most importantly, these obtained density ratio bounds seem to be independent of camera viewpoints and human positions in the image. We have tested the validity of these density ratio bounds and good performance of the online counting of our proposed method on Benchmark of PETS 2009 and some of our video samples. Our system runs in real-time with standard-resolution videos, with average processing rate of around 30 fps.

There are still many rooms for improving this work. We must improve the confidence in large groups of people by conducting more experiments. We can integrate multichannels working simultaneously in the same PC, since we are developing a low-cost solution to counting people. That is, we could count people in different places simultaneously without using extra processing devices.

References

1. Barandiaran, J., Murguia, B., Boto, F.: Real-time people counting using multiple lines. In: WIAMIS, pp. 159–162 (2008)
2. Yu, S., Chen, X., Sun, W., Xie, D.: A robust method for detecting and counting people. In: ICALIP, pp. 1545–1549 (2008)
3. Albiol, A., Mora, I., Naranjo, V.: Real-time high density people counter using morphological tools. *IEEE Trans. ITS* 2, 204–218 (2001)
4. Park, H., Lee, H., Noh, S., Kim, J.: An area-based decision rule for people-counting systems. In: Günsel, B., Jain, A.K., Tekalp, A.M., Sankur, B. (eds.) *MRCSS 2006*. LNCS, vol. 4105, pp. 450–457. Springer, Heidelberg (2006)
5. Haritaoglu, I., Harwood, D., Davis, L.S.: W4: Real-time surveillance of people and their activities. *IEEE Trans. PAMI* 22, 809–830 (2000)
6. Zhao, T., Nevatia, R.: Bayesian human segmentation in crowded situations. In: *CVPR*, pp. 459–466 (2003)
7. Elgammal, A.M., Davis, L.S.: Probabilistic framework for segmenting people under occlusion. In: *ICCV*, pp. 145–152 (2001)

8. Zhao, X., Dellandrea, E., Chen, L.: A people counting system based on face detection and tracking in a video. In: AVSS, pp. 67–72 (2009)
9. Li, M., Zhang, Z.X., Huang, K., Tan, T.N.: Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection. In: ICPR, pp. 1–4 (2008)
10. Kelly, P., O'Connor, N.E., Smeaton, A.F.: Robust pedestrian detection and tracking in crowded scenes. *Image Vision Computing* 27, 1445–1458 (2009)
11. Davies, A.C., Yin, S.A.M., Velastin, S.A.: Crowd monitoring using image processing. *Electron. Commun. Eng. J.*, 37–47 (1995)
12. Marana, A.N., Velastin, S.A., Costa, L.F., Lotufo, R.A.: Automatic estimation of crowd density using texture. *J. Safety Sci.* 28, 165–175 (1998)
13. Kilambi, P., Ribnick, E., Joshi, A.J., Masoud, O., Papanikolopoulos, N.: Estimating pedestrian counts in groups. In: CVIU, vol. 110, pp. 43–59 (2008)
14. Chan, A.B., Morrow, M., Vasconcelos, N.: Analysis of crowded scenes using holistic properties. In: Int'l Workshop on PETS, pp. 101–108 (2009)
15. Albiol, A., Silla, M.J., Albiol, A., Mossi, J.M.: Video analysis using corner motion statistics. In: Int'l Workshop on PETS, pp. 31–37 (2009)
16. Stauffer, C., Grimson, W.L.R.: Learning patterns of activities using real-time tracking. *IEEE Trans. PAMI* 22, 747–757 (2000)
17. Sharma, P.K., Huang, C., Nevatia, R.: Evaluation of people tracking, counting and density estimation in crowded environments. In: Int'l Workshop on PETS, pp. 39–46 (2009)
18. Stalder, S., Grabner, H., Gool, L.V.: Exploring context to learn scene specific object detectors. In: Int'l Workshop on PETS, pp. 63–70 (2009)
19. Cucchiara, R., Grana, C., Piccardi, M., Prati, A., Sirotti, S.: Improving shadow suppression in moving object detection with hsv color information. In: ITSC, pp. 334–339 (2001)

Curved Reflection Symmetry Detection with Self-validation

Jingchen Liu¹ and Yanxi Liu^{1,2}

¹ Department of Computer Science and Engineering

² Department of Electrical Engineering
The Pennsylvania State University
University Park, PA 16802, USA
{jingchen, yanxi}@cse.psu.edu

Abstract. We propose a novel, self-validating approach for detecting curved reflection symmetry patterns from real, unsegmented images. Our method benefits from the observation that any curved symmetry pattern can be approximated by a sequence of piecewise rigid reflection patterns. Pairs of symmetric feature points are first detected (including both inliers and outliers) and treated as ‘particles’. Multiple-hypothesis sampling and pruning are used to sample a smooth path going through inlier particles to recover the curved reflection axis. Our approach generates an explicit supporting region of the curved reflection symmetry, which is further used for intermediate self-validation, making the detection process more robust than prior state-of-the-art algorithms. Experimental results on 200+ images demonstrate the effectiveness and superiority of the proposed approach.

1 Introduction

Symmetry is pervasive in nature and man-made environments [1, 2]. It is one of the most important cues for human and machine perception of the world [1]. Automatic perception of symmetry patterns from images has been a standing research topic in computer vision. Reflection symmetry [2], as one of the four basic symmetries, is the most common and has received most attention in psychology as well as in computer vision [1]. Various applications utilize reflection symmetry such as face analysis [3], multi-target pattern analysis and tracking [4], vehicle detection [5] and medical image analysis [6].

Reflection symmetry detection algorithms dominate the literature of all types of symmetry detections [1, 7]. For example, Sun and Si [8] used histogram of gradient orientations to find the orientation of dominant reflection axis. Masuda, et. al. [9] explored edge features to measure symmetry similarity and Loy and Eklundh [10] matched feature points and then extracted reflection (and also rotation) symmetry patterns via clustering; Mitra et. al. [11] developed partial or approximate Euclidean reflection symmetry detection in subsampled 3D data.

Besides rigid reflection symmetry, Kanade in 1983 proposed the term *skewed symmetry* denoting reflection symmetry of an object going through global affine or



Fig. 1. Some example images containing curved reflection pattern, including real-world/synthesized, segmented/unsegmented, nature/man-made object images

perspective skewing [12]. Symmetry recognition from global affinity and perspective distorted views has also been well studied in [13, 14, 15, 16, 17], where the reflection axis is assumed to be a straight line. In real world however, many symmetrical objects/patterns present curved reflection axes as shown in Figure 1. Automatically recognizing curved symmetry axis from unsegmented images is motivated by a wide range of applications. For example, symmetric region segmentation and curvature analysis from spine x-ray images, as well as leaves recognition and classification, can all benefit from a curved reflection detection algorithm.

Lee and Liu in [18] proposed the first, state-of-the-art curved glide-reflection symmetry extraction algorithm from real, unsegmented images. Their algorithm detects symmetric feature points first, which we refer to in this paper as symmetry ‘particles’ and then clusters these particles in the parameter space, subsequently fits a polynomial function to obtain the curved symmetry axis. The weakness of this approach is the ability against potential outliers (in some cases much more than the number of inliers) contained in the particles, which can seriously affects the robustness of clustering and curve fitting of the reflection axis. Besides, the polynomial fitting of symmetry axis misclassifies many inlier/outlier particles, as a result, the supporting region of the detected symmetry is not well defined thus making it hard to quantitatively assess the reliability of the detected pattern online.

Based on these facts and the abundance of real world curved reflection symmetries (Fig 1), we propose a curved reflection symmetry detection approach that explicitly selects symmetry ‘particles’ with local supporting regions and achieves more robust performance than [18] on curved reflection symmetry extraction by being able to effectively self-validate the detected results.

We adopt the bottom-up framework of [18, 10] that first detects and matches symmetric feature points to form symmetry particles (including both inliers and outliers), while build up symmetric regions in the higher level. The major novel advantage of detecting deformed symmetry patterns from bottom-up is that feature points are free of global deformation, meanwhile local deformation can be

handled by more sophisticated feature points such as SIFT [19], which is robust against scale change and rotation with good repeatability and high efficiency.

A crucial part of our approach is to discover a smooth path going through inlier particles on the image to approximate a valid curved reflection axis. One challenge is that the set of symmetry particles detected in the first step can be misleading. This is because feature point matching only considers local patches around the feature points, and symmetry, on the other hand, is a non-local, continuous feature [20]. There always exist many outlier feature point pairs, that only appear symmetrically in a small local region. To effectively validate the symmetry particles, region-based evaluation and verification are more robust and should be adopted. It can be seen that one symmetry particle is uniquely specified by a pair of feature points, while 2 particles, consisting of 4 feature points, form a closed quadrilateral region. If we approximate the local symmetry axis using straight line within the quadrilateral, a region-based reflection symmetry evaluation step can be done easily and reliably. Therefore given any pair of symmetry particles, we can quantitatively measure the symmetry-ness within the corresponding region, which we refer to as ‘consistency between symmetry particles’, and establish a graph structure with all vertexes representing symmetry particles and the edges representing a straight reflection axis between two particles and the weight on this edge indicating the consistency or local symmetry score. (Figure 2-C,D)

By establishing the graph of linked symmetry particles, we turn this problem of curved symmetry pattern recognition into a problem of seeking a smooth path in the graph that maximize the symmetry property along the path. We will show in Section 2.3 that this is a global optimization problem and we thus propose a multiple hypothesis path sampling and pruning approach for real world curved reflection symmetry detection. Validation results on more than 200 images of three categories show superior curved reflection symmetry detection rates of our algorithm than [18].

One important advantage of explicitly selecting symmetry particles to approximate curved reflection is that we can obtain a well-defined continuous supporting region of reflection symmetry along the curved axis. As we use thin-plate spline (TPS) warping to rectify the axis, we can evaluate quantitatively and globally how symmetric the rectified region is, thus achieving self-validation, making the algorithm more robust.

2 Our Approach

The bottom-up framework for curved reflection symmetry pattern detection starts with recognizing symmetric feature point pairs. Each pair of feature points x_{i1}, x_{i2} provides us with a symmetry axis particle $l_i = \{x_i, \alpha_i\}$, where $x_{i1}, x_{i2}, x_i \in R^2$ are the image coordinates of the two symmetric feature points and their middle point, α_i is the orientation perpendicular to the line joining the points x_{i1} and x_{i2} , with ambiguity of angle π . In the next stage, we evaluate the pairwise consistency among all symmetry particles and establish an undirected graph

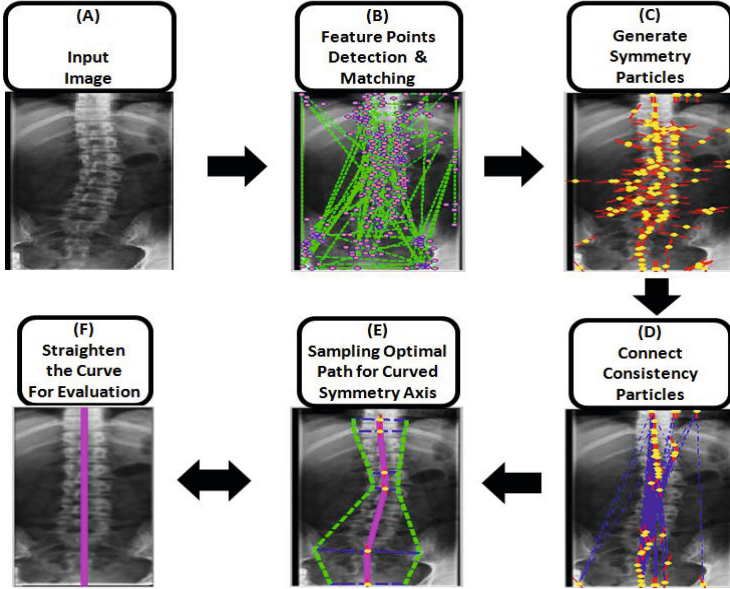


Fig. 2. The framework of our approach: (A)input image; (B)detected SIFT feature points marked as pink dots and successfully matched feature point pairs connected using green dashed lines; (C)representing feature points pairs as yellow particles with red short lines indicating the directions of potential reflection symmetry axis α_i ; (D)Maximally connected components in particle pairwise consistency graph G ; (E)Sampled optimal path from G ; (F)Rectified region via TPS warping

$G\{V, E\}$, with $V = \{l_i\}$ being the set of all particles and any edge $(i, j) \in E$ means the line segment joining particle x_i and x_j reflects the symmetry property locally. The recognition of curved reflection patterns thus becomes a problem of discovering a smooth path from the graph G , which goes through a subset of its vertices (particles), $(l_{i_1}, l_{i_2}, \dots, l_{i_k})$, to approximate the curved reflection axis (Figure 2).

2.1 Symmetry Particles Discovering

We adopt SIFT feature for effective symmetric points recognition since it is rotation and scale-invariant [19]. By rearranging the SIFT descriptor vectors v_i , we can describe the same local patch in the mirror image, denoted by $v_i^{(m)}$. The symmetry distance between two feature points is defined to be the Euclidian distance of the description vectors

$$d(i, j) = \|v_i - v_j^{(m)}\|_2 \quad (1)$$

For each point, we find top 3 best matches with smallest symmetry distance and then reject matches either having different scales or do not satisfy the angular constrains. Let a pair of SIFT feature points be $(x_{i1}, \phi_{i1}, s_{i1})$ and $(x_{i2}, \phi_{i2}, s_{i2})$,

where ϕ_{i1} , ϕ_{i2} and ϕ_{i12} are the orientation angles of two feature points and the line connecting them, respectively. We specify the angular constraint that the orientation of two feature points should also be symmetric, which means $(\phi_{i1} + \phi_{i2})/2 \perp \phi_{i12}$. Each accepted pair of feature points then corresponds to a symmetry particle as illustrated in Figure 3 (a).

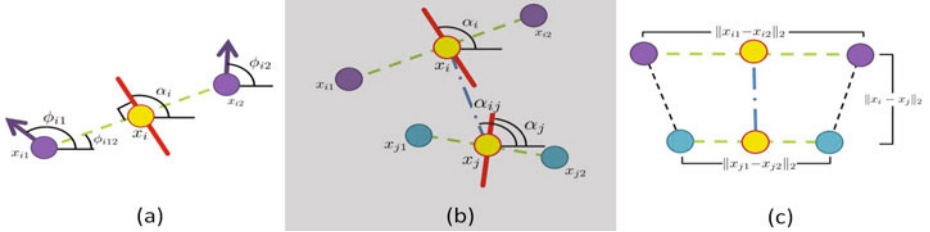


Fig. 3. An illustration of angle constrains and quadrilateral rectification

2.2 Generating Pairwise Consistency Graph

Given two symmetry particles, $l_i = (x_i, \alpha_i)$ and $l_j = (x_j, \alpha_j)$, and let the direction from x_j to x_i be α_{ij} , as illustrated in Figure 3 (b), we evaluate whether they form a near-symmetry region based on both of their geometric consistency and the appearance symmetry score. Assuming α_i and α_j are the tangents of the symmetry axis curve at locations x_i and x_j respectively, the geometric consistency requires that the curve be smooth, which means α_i , α_j and α_{ij} are along the similar directions, leading to the following two conditions:

$$|\alpha_i - \alpha_j| < TH_1 \quad (2)$$

$$|\alpha_{ij} - \frac{1}{2}(\alpha_i + \alpha_j)| < TH_2. \quad (3)$$

In our experiments, we set both thresholds to be $TH = \pi/8$. Once the pair of particles passed the geometric consistency, we rectify the local image patch to evaluate the appearance symmetry score by warping the quadrilateral formed by vertexes $x_{i1}, x_{i2}, x_{j1}, x_{j2}$ into an equilateral trapezoid, where the length of the parallel sides are $\|x_{i1} - x_{i2}\|_2$ and $\|x_{j1} - x_{j2}\|_2$ respectively, and the height is $\|x_i - x_j\|_2$, as illustrated in Figure 3 (c). TPS warping is used in our approach to deal with the most general transformation without assuming any specific cases like affine or perspective.

To evaluate the symmetry score of an equilateral trapezoid, we flip the trapezoid according to the middle axis and calculate normalized cross correlation (NCC) with the original patch, which returns a score between -1 and 1 . If the NCC score is above a threshold (0.5 in our experiments), we impose an edge between particles l_i and l_j and record the NCC score as well as the area (supporting region) of the trapezoid for future use.

2.3 Multiple Hypothesis Path Sampling and Evaluation

The pairwise consistency graph can be further divided into several subgraphs based on connectivity. In cases where multiple symmetry patterns exist, each subset possibly contains one symmetry pattern. In cases of single symmetry pattern detection, we only focus on the subset with maximum number of vertices, which in most cases contains the most dominant pattern.

We then look for a smooth path within the subgraph(s) that maximizes the ‘symmetry’ along it. The symmetry score of a path can be obtained after we use TPS warping to rectify the whole path into connected equilateral trapezoids and evaluate its symmetry score the same way as we do for a single trapezoid. For the sake of computation time, it is also reasonable to approximate the path symmetry score using weighted summation of piecewise scores.

Let a path p going through N vertexes $p = (v_1, \dots, v_i, \dots, v_N)$, or $N - 1$ edges $p = (E_1, \dots, E_i, \dots, E_{N-1})$, with $E_i = (v_i, v_{i+1}, NCC_i, s_i)$, where NCC_i and s_i are the symmetry NCC score and the area of the trapezoid corresponding to the pair of particles l_i, l_{i+1} . We approximate the NCC score of a path using

$$s_p = \sum_{i=1}^{N-1} s_i \quad (4)$$

$$NCC_p = \frac{1}{s_p} \sum_{i=1}^{N-1} s_i \cdot NCC_i \quad (5)$$

To ensure the smoothness of the path, we require the turning angle at each vertex be less than a threshold of $\pi/5$, which means

$$\angle(x_i - x_{i-1}) - \angle(x_{i+1} - x_i) < \pi/5, \quad i = 2, 3, \dots, N - 1, \quad (6)$$

where $\angle()$ is the orientation of a vector, x_i correspond to the 2D coordinate of v_i in the image.

We define 2 criteria c_1 and c_2 given a path p for its ranking, one is the approximation of the path symmetry score, the other also favors paths covering more area:

$$c_1(p) = NCC_p \quad (7)$$

$$c_2(p) = NCC_p + \lambda \cdot \log(S_p) \quad (8)$$

The traditional graph solutions for finding optimal paths such as Dijkstra’s algorithm is not suitable here for the criteria in equations 7 or 8 and the smoothness condition in equation 6, all involves global information of the paths. The enumeration of all possible pathes is also computationally unaffordable. As an alternative, we try to selectively sample the paths with high likelihood.

Once a path is initiated, we can enumerate the next valid vertexes to extend the current path. Each enumeration would generate an extended path hypothesis, each of which can be further extended recursively. Such an iterative approach forms a multiple hypothesis sampling of all possible pathes. The complexity of

this sampling scheme grows exponentially and is unbounded depending on the density of the graph, thus we need to perform efficient pruning to cut unlikely paths in the first place. In each iteration after all current paths having been extended, we prune paths with low likelihood, and only keep a maximum number of K hypothesis in the pool. When $K = 1$, this becomes a greedy algorithm that starts at a random vertex and finds the local optimum; When $K = \infty$, we find global optimum by enumerating all valid paths in the graph that contains the initialization vertex. In our experiment, we take paths ranked top 100 either under criteria 1 or criteria 2. The reason we set up 2 criteria is that although we favor longer curves finally, we want to protect potential paths in the pool before they have been fully extended. This pruning policy effectively bounds the computation within linear complexity meanwhile providing us with good enough solutions.

The algorithm for sampling and pruning paths recursively is illustrated in Table 1, note that once a path p is extended by a new vertex v_{N+1} , its NCC score approximation can be updated also in an efficient recursive form,

$$NCC_p^{(new)} = \frac{NCC_p^{(old)} \cdot s_p^{(old)} + NCC_N \cdot s_N}{s_p^{(old)} + s_N}. \quad (9)$$

For the complete paths (can not be extended any more) returned by the multiple hypothesis sampling, instead of using equation 5 for approximation, the one with highest score according to equation 8 is selected to be the final result.

Algorithm 1. Multiple Hypothesis Path Sampling

Input:

$G(V, E)$, with $V = \{v_i\}$, $E = \{(v_{k1}, v_{k2}, NCC_k, s_k) | k = 1, 2, \dots, K\}$;

Initialize:

Randomly pick a vertex v_i , mark it as incomplete and put it into the path pool;

while exists incomplete paths in the pool

for all incomplete paths in the pool

if the path can be further extended

 replace the path with all valid extend paths;

 update path NCC score (Eq.9) and mark them as incomplete.

else

 Mark the path as complete.

 Prune paths in the path pool with low likelihood score.

Return:

 All paths remaining in the pool.

For the final candidates returned by the multiple-hypothesis sampler, we use criteria c_2 in 8 to rank them, however, instead of using NCC score approximation as equation 5, we use TPS warping to straighten the curved reflection axis to calculate the accurate NCC score. based on the c_2 ranking, the path that produces highest score is selected as final result.

3 Experimental Results and Comparison

We test our algorithm on 210 images including 2 subcategories of the Swedish Leaf dataset [21]—one has curved reflection symmetry pattern on every single leaf (*Quercus rober*, 75 images), the other has curved reflection symmetry pattern among the leaves on a branch (*Sorbus aucuparia*, 75 images), as well as a human-spine X-ray dataset containing 30 images that we collect ourselves. We also collected a set of miscellaneous real-world and synthesized images with curved reflection symmetry patterns (30 images). In addition, we compare our approach with the method in [18] on the same image sets.

3.1 Our Results

A representative selection of our results on the Swedish leaf dataset, spine X-ray dataset and miscellaneous images is shown in Figures 4, 5, 6¹. For each image, we tag the detection results of curved reflection axis, as well as its support region specified by symmetric feature points. We also straighten the curved reflection axis and show the rectified image in the supporting region on the right.

(H) and (I) in Figure 6 show 2 different reflection patterns being detected from the same image. This is achieved when we separate the consistency graph G into several connected components, each of which could be checked for the existence of reflection patterns.

It can be seen from these results and rectified images that piece-wise rigid reflection is a reasonable approximation for curved reflection. Our method is effective and robust in selecting a small subset of inlier symmetry particles explicitly to represent the reflection pattern. If necessary, curve fitting can be further applied on the selected inlier particles to obtain a smoother curve.

Although we introduced several heuristic thresholds in the algorithm, e.g., $TH_1 = TH_2 = \pi/8$, $TH_{NCC} = 0.5$, they are mainly for efficiency concerns. A stricter threshold helps saving time by pruning bad hypothesis in an earlier stage; Relaxing the thresholds would result to more outliers being included in the path-sampling stage. However the final results are relatively insensitive against the threshold changes, which is because our approach has the self-validation ability, making it perform robustly in finding the correct inlier particles even when the outliers are more than inliers (which is usually the case).

In addition, we make some extra tests on images containing human perceived straight-reflection symmetries (Figure 7). The results demonstrate a more accurate capture of the slight deformations of the seemingly straight reflection axes by our piece-wise curve approximation algorithm, indicating the rarity of perfectly straight reflection symmetries in real world.

Some failure examples are also shown in Figure 8, where (a) failed due to severe background clutter; (b) and (c) failed because not enough feature points are detected in the first place so that the dominant reflection symmetry pattern

¹ See our project page for a complete set of Data and Results at <http://vision.cse.psu.edu/research/curvedSym/index.shtml>

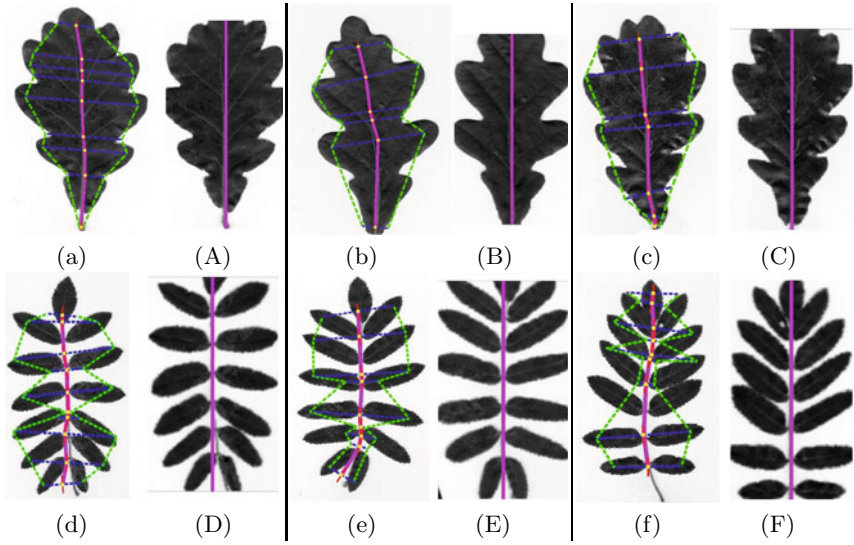


Fig. 4. Results of Swedish leaf data set. top row: single leaf with curved reflection symmetry pattern; bottom row: multiple leaves form curved reflection symmetry pattern; a-f: original images tagged with detected curved reflection axis(pink), supporting region(green); A-F: rectified images with a straightened reflection axis(pink).

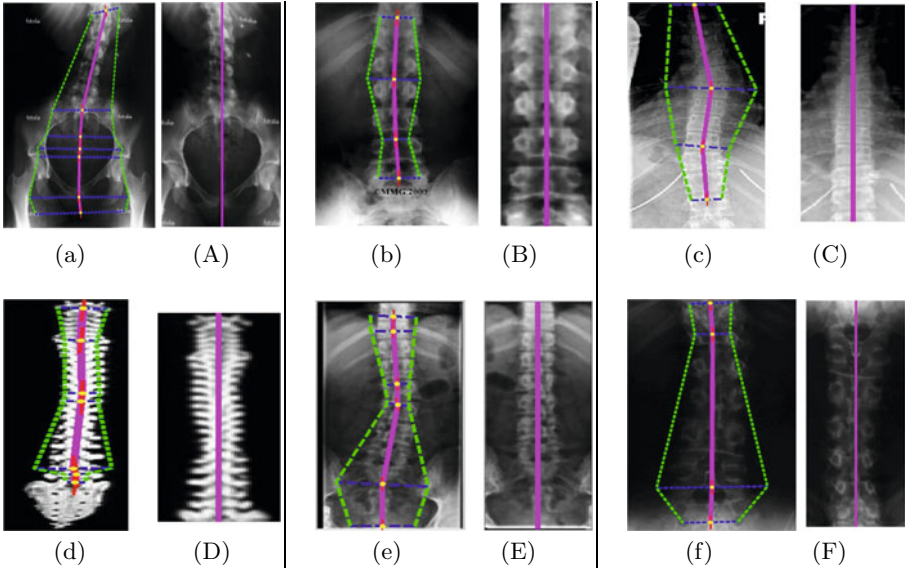


Fig. 5. Results of Spine X-ray data set. a-f: original images tagged with detected curved reflection axis(pink), supporting region(green); A-F: rectified images with a straightened reflection axis(pink).

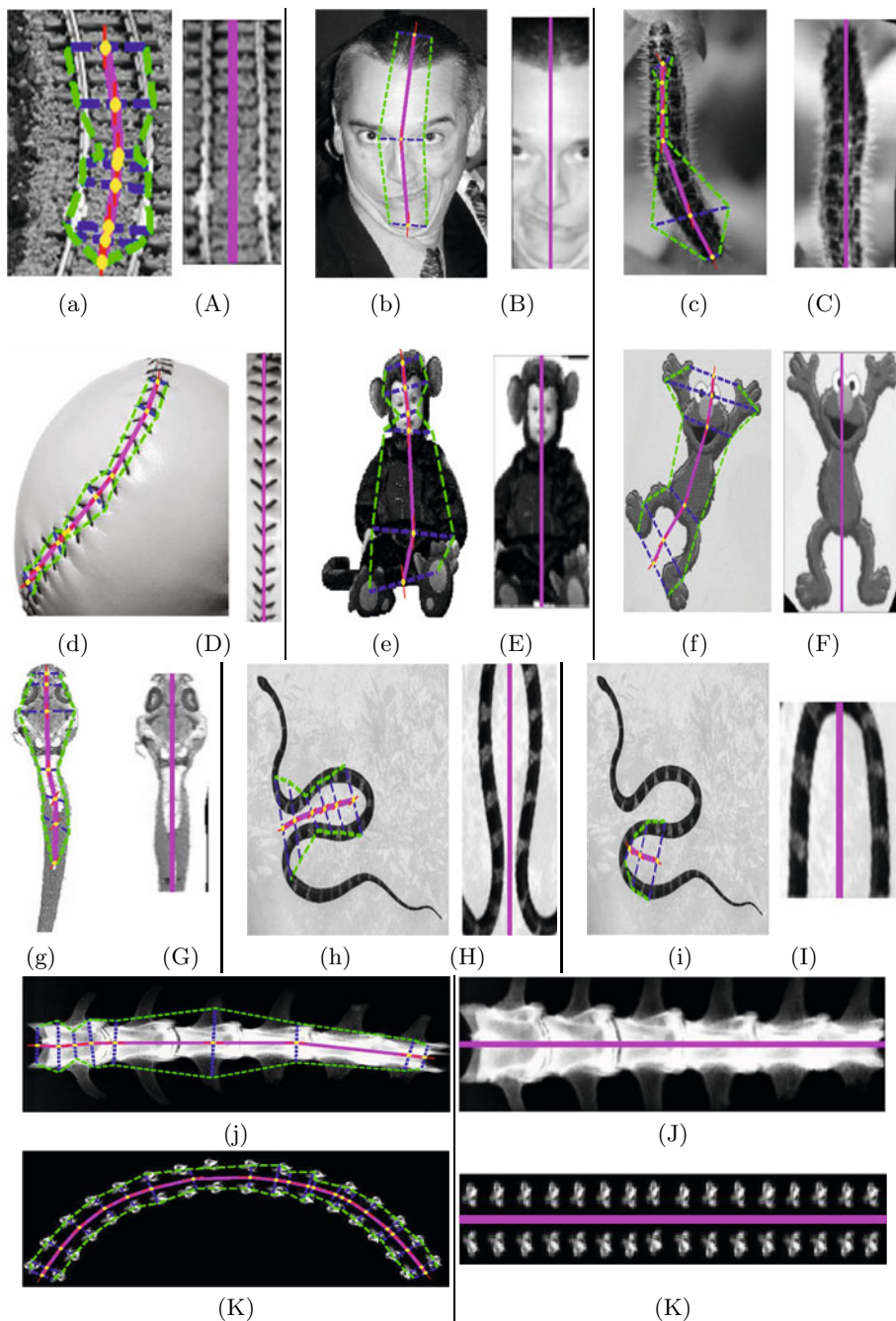


Fig. 6. Results of miscellaneous images. a-k: original images tagged with detected curved reflection axis(pink), supporting region(green); A-K: rectified images with a straightened reflection axis(pink).

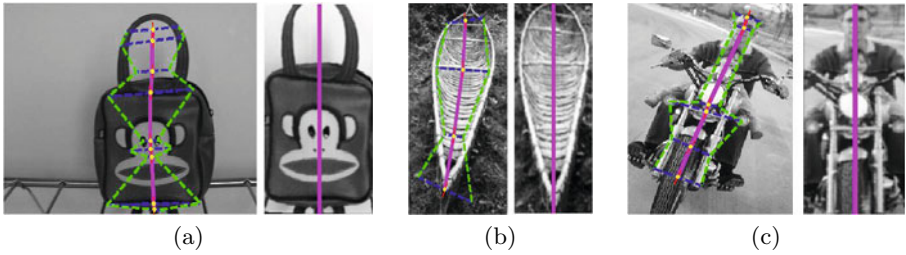


Fig. 7. Examples of detecting almost-straight reflection symmetries

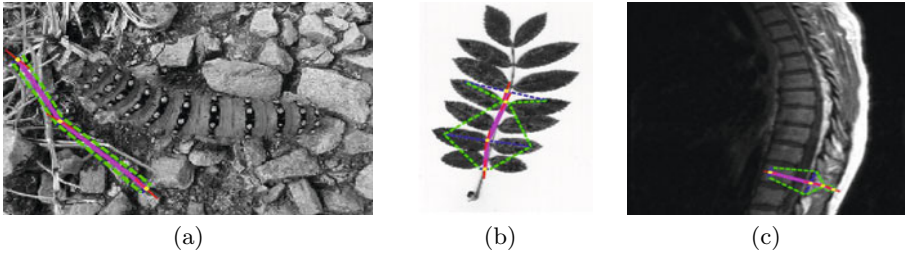


Fig. 8. Examples of failed cases, where (a) failed due to severe background clutter; (b) and (c) failed because not enough feature points are detected in the first place thus missed the dominant reflection pattern, while finding some local reflection symmetries

is not (fully) recognized. From the experiment, most of the failure cases are due to not enough feature points being extracted. Therefore to make our approach more robust, multiple types of feature point detection can be adopted here e.g., Harris-Laplace [22], which detects corner like points and is complement with Hessian-Laplace (blob-like) feature points.

3.2 Quantitative Evaluation and Comparison with [18]

We also apply Lee and Liu’s approach [18] to the same data sets and make quantitative comparisons. For each image, we tag it success if more than 4/5 of the curved reflection axis is detected, and failure otherwise. Our method has higher success rate on all three datasets as reported in Table 1.

Some of the detection results of Lee and Liu’s in [18] are also shown here in Figure 9 for an intuitive comparison. It can be seen that by defining an

Table 1. Success rates of our proposed algorithm and Lee & Liu’s [18]

Dataset	Leaf dataset	Spine dataset	Miscellaneous images	Overall
# images	150	30	30	210
proposed	83.3%	80.0%	73.3%	81.4%
Lee & Liu [18]	40.0%	66.7%	70.0%	48.1%

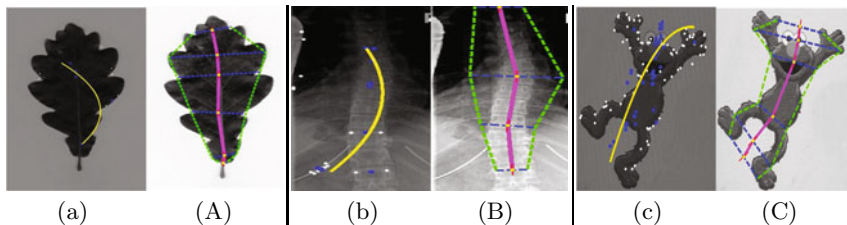


Fig. 9. Comparisons of our proposed approach with Lee & Liu in [18]. a-c: Lee & Liu’s approach; A-C: our approach.

explicit supporting region and TPS warping to rectify the curve, we achieve self-validation in our method thus being more robust against outliers and yield to better performance.

4 Conclusions

In this paper, we propose a bottom-up curved-reflection symmetry detection approach, starting from recognizing symmetric points pairs (particles) in the bottom level and extract a consistent structure among the particles to form the symmetry pattern in the higher level. Multiple-hypothesis sampling and pruning method is shown to be effective in discovering the optimal curved structures from real world images. As a by-product, we obtain the supporting regions from selected particles and use them for self-validation. Quantitative evaluation and comparison against state-of-the-art algorithm on 210 real images confirm the superior robustness of our proposed approach.

Acknowledgement. We thank Lee and Liu [18] for providing their source code. This work is supported in part by an NSF grant IIS-0729363 and a gift grant to Dr. Liu from Northrop Grumman Corporation.

References

1. Liu, Y., Hel-Or, H., Kaplan, C.S., Gool, L.V.: Computational symmetry in computer vision and computer graphics. *Foundations and Trends in Computer Graphics and Vision* 5, 1–195 (2010)
2. Weyl, H.: *Symmetry*. Princeton University Press, Princeton (1952)
3. Mitra, S., Liu, Y.: Local facial asymmetry for expression classification. In: *Proceedings of the 2004 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2004)*, vol. 2, pp. 889–894 (2004)
4. Liu, J., Liu, Y.: Multi-target tracking of time-varying spatial patterns. In: *Computer Vision and Pattern Recognition Conference, CVPR 2010* (2010)
5. Kuehnle, A.: Symmetry-based recognition of vehicle rears. *Pattern Recogn. Lett.* 12, 249–258 (1991)

6. Mancas, M., Gosselin, B., Macq, B.: Fast and automatic tumoral area localisation using symmetry. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP 2005), vol. 2, pp. 725–728 (2005)
7. Park, M., Lee, S., Chen, P.C., Kashyap, S., Butt, A.A., Liu, Y.: Performance evaluation of state-of-the-art discrete symmetry detection algorithms. In: Computer Vision and Pattern Recognition Conference (CVPR), pp. 1–8 (2008)
8. Sun, C., Si, D.: Fast reflectional symmetry detection using orientation histograms. *Real-Time Imaging* 5, 63–74 (1999)
9. Masuda, T., Yamamoto, K., Yamada, H.: Detection of partial symmetry using correlation with rotated-reflected images. *Pattern Recognition* 26 (1993)
10. Loy, G., Eklundh, J.: Detecting symmetry and symmetric constellations of features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 508–521. Springer, Heidelberg (2006)
11. Mitra, N., Guibas, L., Pauly, M.: Partial and approximate symmetry detection for 3d geometry. *ACM Transactions on Graphics* 25, 560–568 (2006)
12. Kanade, T., Kender, J.R.: Mapping image properties into shape constraints: skewed symmetry, affine-transformable patterns, and the shape-from-texture paradigm. In: *Human and Machine Vision*, pp. 237–257 (1983)
13. Shen, D., Ip, H., Teoh, E.: Robust detection of skewed symmetries. In: *International Conference on Pattern Recognition*, vol. 3, pp. 1010–1013 (2000)
14. Van Gool, L., Proesmans, M., Moons, T.: Mirror and point symmetry under perspective skewing. In: *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 285–292 (1996)
15. Carlsson, S.: Symmetry in perspective. In: Burkhardt, H.-J., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1406, p. 249. Springer, Heidelberg (1998)
16. Lei, Y., Wong, K.: Detection and localisation of reflectional and rotational symmetry under weak perspective projection. *Pattern Recognition* 32, 167–180 (1999)
17. Cornelius, H., Loy, G.: Detecting bilateral symmetry in perspective. In: *CVPRW*, p. 191 (2006)
18. Lee, S., Liu, Y.: Curved glide-reflection symmetry detection. In: *Computer Vision and Pattern Recognition Conference (CVPR 2009)*, pp. 1046–1053 (2009)
19. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60, 91–110 (2004)
20. Zabrodsky, H., Peleg, S., Avnir, D.: Symmetry as a continuous feature. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17, 1154–1166 (1995)
21. Söderkvist, O.J.O.: Computer vision classification of leaves from swedish trees. Master's thesis, Linköping University (2001)
22. Mikolajczyk, K., Schmid, C.: Scale & affine invariant interest point detectors. *International Journal of Computer Vision (IJCV)* 1, 63–86 (2004)

An HMM-SVM-Based Automatic Image Annotation Approach

Yinjie Lei, Wilson Wong, Wei Liu, and Mohammed Bennamoun

School of Computer Science and Software Engineering
University of Western Australia
35 Stirling Highway, Crawley WA 6009
{yinjie,wilson,wei,bennamou}@csse.uwa.edu.au

Abstract. This paper presents a novel approach to **Automatic Image Annotation (AIA)** which combines both **Hidden Markov Model (HMM)** and **Support Vector Machine (SVM)**. Typical image annotation methods directly map low-level features to high-level concepts and overlook the importance to mining the contextual information among the annotated keywords. The proposed HMM-SVM based approach comprises two different kinds of HMMs based on image color and texture features as the first-stage mapping scheme and an SVM which is based on the prediction results from the two HMMs as a so-called high-level classifier for final keywording. Our proposed approach assigns 1-5 keywords to each testing image. Using the Corel image dataset, Our experiments have shown that the combination of a discriminative classification and a generative model is beneficial in image annotation

1 Introduction

The modern developments of the Internet make it the most efficient platform for obtaining and sharing various kinds of information from anywhere. For this reason the research into search engines for retrieving and managing multimedia data has become very important and attractive [1]. Existing search engines are well-developed in the case of textual data. However, more research is still required for image search and retrieval due to the so-called *semantic-gap*. At the early stage of research, image retrieval was performed by relying on manually assigned keywords. The manual labeling of images however is tedious and difficult for large image collections. To address these drawbacks, content-based image retrieval using low-level image features such as color, texture and shape is proposed [2]. These low-level features representing visual content of an image can be used to measure the similarity between images. This allows images from datasets to be automatically indexed and searched. To improve the process of retrieval, this line of research based on low-level features was soon replaced by the use of the approach of AIA which associates multiple keywords with objects in images. Some researchers argued that if we can associate multiple keywords with the identified object in the image, the retrieval of images could become much easier and more straightforward [3,4,5,6].

For this reason, AIA has become a focus in the area of content-based image retrieval to bridge the semantic gap [7]. In recent years, the classifier ensembles approach has attracted much more attention. Some results report that it is more reliable than most one-level classifier in performing automatic image annotation [8]. Another trend of classification is the fusion with other techniques to enhance the performance [9]. In practice, the intrinsic advantages of generative model have been widely accepted and used in the area of automatic image annotation. Recently, one representation of generative models, namely the Hidden Markov Model has been utilized to resolve automatic image annotation problems [10]. However, there are still opportunities to improve the quality of automatic image annotation for two reasons. First, images which are semantically similar often contain different low-level features. Therefore the direct mapping of the low-level features to high-level concepts may lead to errors. Second, most existing approaches overlook the significance of keyword correlation in image retrieval. For instance, ‘boat’ and ‘water’ tend to co-occur much more often in one image than ‘boat’ and ‘grass’. This suggests the correlation information among keywords can be of great help to improve the performance of AIA.

In this paper, we present a two-stage mapping AIA technique based on both Support Vector Machine and Hidden Markov Model. The first stage comprises two HMMs constructed separately from color and texture features of images for mapping the low-level features to mid-level features. Co-occurrence based keyword correlation is also constructed to enhance the mapping precision. In the second stage we employ support vector machine to map the so-called mid-level features to high-level concepts. The proposed scheme fuses both a discriminative classification and a generative model to avoid the two problems discussed above.

The outline of this paper is as follows: Recent image annotation methods based on both SVM and HMM are briefly reviewed in Section 2. Our proposed SVM-HMM based annotation approach is explained in Section 3. Section 4 presents the experimental results and the performance analysis of the proposed method. The paper is then concluded in Section 5.

2 Related Work

Automatic image annotation techniques first appeared about two decades ago. Below is a review of some selected milestones in AIA using SVM and HMM.

Support Vector Machine was first introduced into this area during the last decade. As a very strong data mining technique, one of the first SVM based image classification system paper is [11]. However they only use global color features to solve a small scale classification problem. With the aim to improve the classification accuracy based on a single classifier, a sophisticated classifier system called “classifier ensembles” was introduced to further improve AIA precision. Gao et al. [12] use a combination of multiple SVM classifiers. These classifiers are obtained by combining the output of several effective weak classifiers using a

Boosting technique. Subsequently, Qi and Han [13] also use a combination of two sets of SVMs which relies on the regional image features found using Multiple Instance Learning (MIL) and global image features respectively. Tsai et al. [8] present an image indexing and classification system called CLAIRE. Their system is based on a Two Stage Mapping Model (TSM) [14]. In their system, three SVMs are constructed as low-level feature classifiers focusing on classifying color and texture features respectively. Another SVM called high-level classifier is constructed based on the outputs of the first low-level classifiers. This system avoids the direct mapping of the low-level features to high-level concepts, and the results show a promising way to assign keywords to images.

As one representative work of generative models, HMM has also been adopted by some researchers to perform AIA. In [15], a one-dimensional hidden Markov model (HMM) was trained on vector-quantized color histograms of image blocks. However, their system can only be used to solve a binary image classification problem. Li and Wang [16] proposed a system called ALIP which is based on a two-dimensional multi-resolution HMM fed by regional image features. Modestino and Zhang [17] use a Markov random field model to capture the spatial relationships between regions and apply a maximum posteriori rule to interpret images. Ghoshal et al. [10] use an HMM for image and video annotation based on two datasets individually, which are COREL and TRECVID. A novel TSVM-HMM based annotation scheme is proposed in [18]. Compared with previous annotation methods, the proposed TSVM-HMM based annotation scheme can achieve better annotation performance with less labeled training images as demonstrated.

3 Proposed Approach

In order to overcome the problems discussed above, we propose a Two-Stage Mapping Model to perform Automatic Image Annotation (AIA). An overview of the proposed approach is shown in Fig.1. The first module is composed of two Hidden Markov Models which are responsible for classifying low-level color and texture features respectively. The second module is an SVM classifier which serves as a high-level classifier as in the work of [8] to determine the final annotation results. Unlike that paper, our approach substitutes the SVM with an HMM during the first stage aiming at mining keyword correlations. Meanwhile, we directly use the category names to define the output of the HMMs. This is to overcome the difficulty of only using twelve colors to describe a large number of images. Moreover, all the image regions of our training set are used as opposed to using only the central region. Therefore the image regions used in our approach may contain different objects. This change also adds difficulty to the process of describing image regions with only twelve color names. Once these two stages are completed, five keywords corresponding to the five sub-blocks can automatically be assigned to a test image. Below is a description of each module of our approach in Fig. 1.

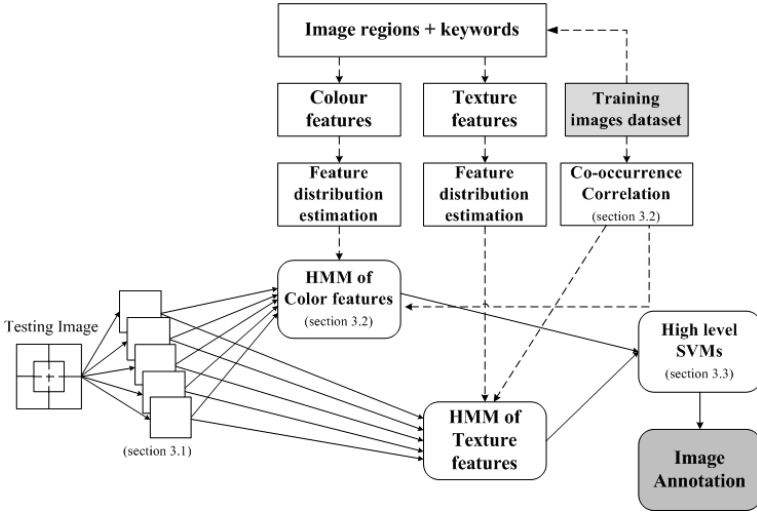


Fig. 1. A block diagram of our proposed approach

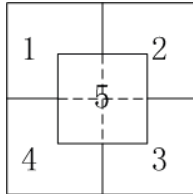


Fig. 2. The tilting scheme

3.1 Image Sub-blocking and Feature Extraction

It is well known that automatic image segmentation is a hard task and no approach can achieve perfect results. Moreover, some results show that those models using a sub-blocking scheme perform better than those using object-based segmentation [19]. We use a tilt scheme which was proposed in [8] to divide images into five regions. The original image size in our dataset has 384×256 pixel resolution, and the region size is 192×128 as shown in Fig.2. The regions include four quadrants. The one in the center is used to increase the weight of the object of interest.

We extract color features and texture features as image descriptors. We do not consider other features such as shape for two reasons. First it is well known that image shape feature extraction is difficult to achieve and computationally expensive. In addition it should be noted that image regions whether containing homogeneous objects or not is not a focus in our approach. Therefore it is meaningless even if image shape feature extraction is applied.

The color features include the mean and standard deviation of every region in the RGB and Lab color spaces. It has been proved that Garbor filter performs

well on extracting image texture features. Therefore we apply a set of Garbor filters with 12 orientations (i.e. $0^\circ, 30^\circ, 60^\circ, \dots, 270^\circ$) on the luminance component of image regions. We then extract the mean and standard deviation values of the 12 filtered images and use them as texture features. This results in a feature vector of length 36 for each region (i.e. 12 color features and 24 texture features).

3.2 Hidden Markov Model for Low-Level Annotation

Hidden Markov Model for AIA. According to HMM's definition, it is easy to provide a density function to model image features of image regions which belong to the same keyword. By introducing keyword correlation, the context-dependent HMM can improve its accuracy for image annotation.

For the sake of brevity, let $T_i = \{I_{i1}, I_{i2}, \dots, I_{in}\}$ be the feature set of image regions obtained from our training set for the i th keyword, where n is the total number of image regions. The keyword set $K = \{k_1, k_2, \dots, k_i\}$ represents all the keywords appearing in the whole training set. Given an image, it will be divided into five regions as described above, where the regions are ordered according to the quadrants as shown in Fig. 2. The upper-left one is considered as the first while the center one as the last. Meanwhile, we use $I_r = \{I_{r1}, I_{r2}, \dots, I_{r5}\}$ to denote its region feature set and $I_c = \{I_{k1}, I_{k2}, \dots, I_{k5}\}$ to denote its keyword set. We propose to model the AIA task as a Hidden Markov process. Thus, by combining I_r and I_c , the joint likelihood function can be formulated as

$$f(I_{r1}, I_{r2}, \dots, I_{r5}, I_{k1}, I_{k2}, \dots, I_{k5} | k_0) = \sum_{k_t \in I_k} \prod_{t=1}^5 f(I_t | k_t) p(k_t | k_{t-1}) \quad (1)$$

According to Eq. (1), an HMM model is mainly affected by the emission density function f . This function corresponds to the image region feature distribution of one keyword. The transition probability function p on the other hand reflects the keyword correlations. The problem then becomes how can we formulate the emission density function and transition probability function.

Low-level Feature Distribution. As discussed in the last subsection, we should establish a useful emission density function for each keyword [10]. Gaussian Mixture Model (GMM) is one of the most statistically mature methods for density estimation [18]. GMM is the weighted average of Gaussians, and each Gaussian has its own mean and covariance matrix which has to be estimated separately. In the proposed approach, we use GMM to model the low-level image features via relevant image regions. Let $T_c = \{I_{c1}, I_{c2}, \dots, I_{cn}\}$ and $T_t = \{I_{t1}, I_{t2}, \dots, I_{tn}\}$ denote the extracted color and texture features of image regions assigned with keyword k . We then employ a Gaussian mixture model with three components to construct the color and texture feature distribution functions $f_c(T_c|c)$ and $f_t(T_t|c)$ as follow,

$$f_{c,t}(T_{c,t}|k) = \alpha_1 g(T_{c,t}; \mu_1, \Sigma_1) + \alpha_2 g(T_{c,t}; \mu_2, \Sigma_2) + \alpha_3 g(T_{c,t}; \mu_3, \Sigma_3) \quad (2)$$

$$g(T_{c,t}; \mu_i, \Sigma_i) = \frac{1}{\sqrt{(2\pi)^{d_{c,t}} |\Sigma_i|}} \exp\left[-\frac{1}{2}(T_{c,t} - \mu_i)^T \Sigma_i^{-1} (T_{c,t} - \mu_i)\right] \quad (3)$$

where $\alpha_1, \alpha_2, \alpha_3$ represent the weight of each Gaussian component respectively, and $\alpha_1 + \alpha_2 + \alpha_3 = 1$. μ and Σ denote the mean and covariance matrix respectively. d denotes the dimension of an image region feature. Here $d_c = 12$ for color features and $d_t = 24$ for texture features.

Keyword Correlation. The transition probability $p(k_t|k_{t-1})$ reflects the correlation between the keywords k_t and k_{t-1} . Here we use k_i and k_j to replace k_t and k_{t-1} respectively. The keyword correlation $p(k_i|k_j)$ is measured by counting the frequency of paired words assigned to each image. We can estimate conditional and joint probabilities of p if we take: $p(k_i|k_j) = \frac{p(k_i, k_j)}{p(k_j)}$, and $p(k_i, k_j) = \frac{N(k_i, k_j)}{|D|}$, where $N(k_i, k_j)$ indicates the number of times k_i and k_j appear together in one image, and $|D|$ is the total number of image regions in the training set. If we repeat this process for each pair of words in the keyword set we can obtain an $i \times i$ conditional probability matrix P_M , which reflects the keyword correlation.

A problem with the P_M matrix is that some of the keywords never appear in the same image. Thus some $p(k_i, k_j)$ may take a value of zero. We apply a widely used smoothing technique known as ‘‘interpolation smoothing’’ to solve this problem. It can be summarized by Eq. 4.

$$p(k_i|k_j) = \beta * \frac{N(k_i, k_j)}{N(k_j)} + (1 - \beta) * \frac{N(k_j)}{|D|} \quad (4)$$

where β is an interpolation parameter and $|D|$ is the number of words in the collection. This formula is an interpolation between the empirical estimate $\frac{N(k_i, k_j)}{N(k_j)}$ and the empirical distribution of the term k_j . Therefore even if two keywords never appear together, we will not have a zero value in P_M . It should be noted that both the color and texture HMMs share the same P_M . It is easy to understand that although the feature sets are different, they should have the same keyword correlation.

Predictions of HMM. The objective of AIA is to find the optimal hidden keyword sequence for regions with learnt HMM. Once the density estimation of $f_{c,t}(T_{c,t}|k)$ for color and texture features of all keywords and transition probabilities have been estimated, given a test image, we perform the Balm-Welch algorithm to compute the posterior probability of each prediction as the first-stage annotation. The posterior probability $d_j(I_t)$ of being predicted with k_j is iteratively achieved using:

$$d_j(I_t) = f(I_t|k_j) \sum_{i=1}^M d_i(I_{t-1}) p(k_i|k_j) \quad (5)$$

In Eq. 5, the posterior probabilities of M keywords, i.e. $d_j(I_t), j = 1, \dots, M$, are acquired through the association with a visible region I_t . The color and texture predictions $j'_{c,t}$ of the hidden keyword for the region I_t can be gained based on the following criterion:

$$j' = \operatorname{argmax}_j(d_j(I_t)) \quad (6)$$

3.3 High-Level Concept Classifier

Training Set for High-level concept classifier. Unlike [8], our approach directly extracts the predictions from the colour and texture Hidden Markov Model. Let $T_i = \{I_{i1}, I_{i2}, \dots, I_{in}\}$ be all the feature set of one keyword. We then apply the constructed color HMMs and textures HMMs to the set and collect the color and texture predictions. After this process, we collect the prediction set $T_i = \{(c_{i1}, t_{i1}), (c_{i2}, t_{i2}), \dots, (c_{in}, t_{in})\}$.

Let us take the concept 'grass' as an example. After applying the HMMs to the training set, we collect all predictions belonging to the image regions which are labelled 'grass'. Let $T_{grass} = \{I_{grass1}, I_{grass2}, \dots, I_{grassN}\}$ be all the image regions under the keyword 'grass'. We assume the color and texture predictions for I_{grass1} as (*tree - color, grass - texture*) and for I_{grass2} as (*grass-color, sky-texture*). If we repeat the application of the HMMs for all the image regions belonging to the keyword 'grass', we can collect all the predictions for that keyword. Then we use this kind of predictions as mid-level features for each keyword. Let M_{grass} denotes the output of the HMM for the keyword 'grass', and $M_{grass} = \{(tree - color, grass - texture), (grass - color, sky - texture), \dots\}$.

In fact, every image region for all the keywords is mapped into a space that we call the HMM prediction space. This space maps color feature predictions on the X-axis and texture feature predictions on Y-axis. According to the total number of keywords over all the training set, which is 120, both the X- and Y-axis will take values ranging from 1 to 120. Fig. 3 shows an example of an HMM prediction space.

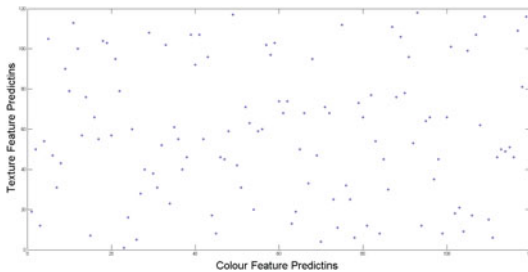


Fig. 3. An example of HMM prediction space to illustrate the use of SVMs for learning predictions based on different keywords

High-level concept SVM. For classifier design, support vector machines (SVMs) are chosen for the image classification task because of their generalization performance superiority. For the selection of the kernel function for the nonlinear mapping, a degree-2 polynomial kernel is used. We use the predictions of HMMs as input to the SVM, and when the number of keywords is i , therefore this method constructs i SVM classifiers. Similar to the multi class classification employed in this paper, the one-against-all method is used. Each input is classified into one positive (+1) and $C - 1$ negative (-1) classes.

3.4 Summary of the Training Procedure

- *Step 1.* Given training set image regions for all keywords. These training sets are composed of 12 color features and 24 texture features.
- *Step 2.* Use the extracted color and texture features to construct the color and texture feature distribution functions f_c and f_t used as density estimates for HMMs as shown in Equations 2 and 3.
- *Step 3.* Investigate the keyword correlation based on the labelled regions to obtain the co-occurrence matrix P_M as transition probability function as described in Equation 4.
- *Step 4.* Return to the training sets in Step 1. Use the constructed HMMs to collect the color and texture predictions of all the image regions. By this time, the training set belonging to each keyword would have been mapped to the prediction space. At the end of this step, the first mapping stage would have been generated.
- *Step 5.* Use the prediction space of each keyword to obtain the high-level concept SVMs. At the end of this step, the second mapping stage would have been generated.

4 Experiments

4.1 Dataset

We tested the proposed AIA approach on the Corel dataset with 5600 images. A selection of 3456 images in the dataset was initially divided into five regions, and all regions were grouped into 120 keywords. Since a region may contain different objects, if one object occupies more than half in the region, the object name will be assigned to this region. We also discarded some regions which are difficult to label. During the training process, every keyword contained around 54 to 810 regions, with a total of 13754 training regions. Next, another 635 images which had no regions appeared in the training set. They were randomly chosen from the dataset and used as testing images. The proposed approach comprises one color, one texture HMMs and 120 SVMs. The color and texture names are the same as the 120 keywords predicted using our HMMs. During the testing process, 5 keywords were automatically assigned by the proposed approach to the testing images.



Fig. 4. Some region samples of four keywords used for training

Table 1. Performance comparison with other methods in terms of average precision and recall for all keywords

	HMM	CLARIE	HMM-SVM
Words with recall > 0	59	0.76	102
Average words recall	0.21	0.34	0.47
Average word precision	0.19	0.32	0.45

4.2 Comparison with Other Methods

The contribution of the proposed HMM-SVM based annotation scheme is to integrate both the discriminative classification and the generative model so as to take full advantage of their combined merits. To evaluate its effectiveness, we compared our HMM-SVM based approach with other two related approaches, namely CLAIRE [8] and HMM-based image annotation [10]. For each method, we assess the annotation performance using the average precision and recall, over all testing images. The precision and recall values are defined in Eq. (7):

$$precision(c) = \frac{num_c}{num_{ca}}; \quad recall(c) = \frac{num_c}{num_{cm}} \quad (7)$$

where num_c denotes the number of image correctly annotated with keyword c , num_{ca} denotes the number of images automatically annotated with keyword c and num_{cm} denotes the number of images manually annotated with keyword c .

Table 1 shows the average annotation precision and recall over the total 120 keywords. Clearly, we can see that the proposed HMM-SVM based annotation method achieves a significant improvement on our experimental dataset. Compared to the other two methods, it shows an improvement of about 26% and 0.13% in recall and 26% and 13% in precision. Moreover, the number of keywords with positive recalls has increased by 43 and 26. Fig. 5 presents some examples of the annotations produced by the proposed approach. The potential reasons for this improvement can be associated to the following: (1) with the




	Ground truth Annotation	HMM-SVM Annotation
	Sky, sky, grass people church	Sky, sky, grass, temple, people
	Sky, mountain, grass, grass, elephant	Sky, sky, grass, grass, elephant
	Cloud, cloud, water, water, castle	Cloud, cloud, water, water, castle

Fig. 5. Examples of image annotation

two-stage mapping scheme involved, it is believed that HMM-SVM can outperform those use only single-mapping approaches such as HMM method; (2) the enhanced keyword correlation is also introduced into the proposed AIA approach and hence keyword semantics is capable of being modelled well compared to other methods that do not consider such correlation (e.g. CLARIE). More details are provided in the next section.

4.3 Effectiveness of the Proposed Approach

To construct a reliable generative model, i.e. HMM, our approach employs a keyword correlation with an interpolation smoothing technique and further promotes the performance of HMM. Three schemes are used to obtain the keyword correlation, i.e. the co-occurrence based keyword correlation, the co-occurrence based keyword correlation without interpolation smoothing, and without keyword correlation which is set to be uniform. As shown in Fig. 6, the performance of annotation is greatly improved when taking into consideration keyword correlations, a concept not used in previous annotation approaches. By combining

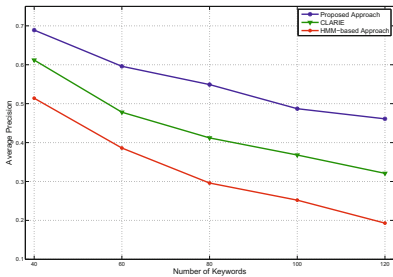


Fig. 6. Evaluation of the effectiveness of keyword correlation based on three schemes described in section 4.3

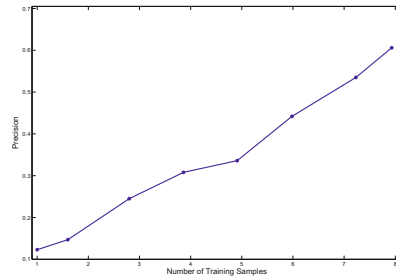


Fig. 7. The annotation precision of eight keywords which contain different number of training region samples

co-occurrence correlation measurements and the interpolation smoothing technique, it can provide more reliable keyword correlation to avoid zero values.

We further examine the relationship between the precision results and the number of training samples. The ten keyword samples randomly selected from the training set, as in Fig. 7, shows an approximately linear relationship between the number of training samples and the annotation precision. We see that a larger number of training samples is a major factor for a better annotation. Therefore, performance should be heavily dependent on the low-level feature representations which employ visual feature distribution functions.

5 Conclusion

In this paper, we proposed an approach for Automatic Image Annotation based on the concept of two-stage mapping. Unlike existing two-stage mapping models, the proposed approach combines the advantages of two-stage mapping and keyword correlation. This two-stage mapping scheme avoids the direct mapping of low-level features to high-level concepts. The keyword correlation mechanism is able to capture to a certain extent the meaning of words to improve the performance of AIA. Our experimental results using the Corel image dataset show that, in the case of annotating images with few words, the combination of the discriminative classification and the generative model can improve annotation performance. Thus, the combination of HMM and SVM provides a promising way to perform and improve automatic annotation of images.

Acknowledgement. This research was supported by the Australia Research Council grant and the University of Western Australian and the China Scholarship Council Joint Scholarship.

References

1. Li, J., Wang, J.: Real-time computerized annotation of pictures. In: Proceedings of the ACM Multimedia Conference, pp. 911–920 (2006)
2. Smeulders, A., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 1349–1380 (2000)
3. Duygulu, P., Barnard, K., de Freitas, N., Forsyth, D.: Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) *ECCV 2002*. LNCS, vol. 2353, pp. 97–112. Springer, Heidelberg (2002)
4. Barnard, K., Duygulu, P., Forsyth, D., de Freitas, N., Blei, D., Jordan, M.: Matching words and pictures. *Journal of Machine Learning Research* 3, 1107–1135 (2003)
5. Blei, D., Jordan, D.: Modeling annotated data. In: 26th Annual International ACM SIGIR Conference, pp. 127–134 (2003)
6. Jeon, J., Lavrenko, V., Manmatha, R.: Automatic image annotation and retrieval using cross-media relevance models. In: 26th Annual International ACM SIGIR Conference, pp. 119–126 (2003)

7. Rui, Y., Huang, T., Chang, S.: Image retrieval current techniques, promising directions and open issues. *J. Visual Commun. Image Representation* 10, 39–62 (1999)
8. Tsai, C., McGarry, K., Tait, J.: Claire: A modular support vector image indexing and classification system. *ACM Transactions on Information Systems* 24, 353–379 (2006)
9. Wong, W., Hsu, S.: Application of svm and ann for image retrieval. *European Journal of Operational Research* 173, 938–950 (2006)
10. Ghoshal, A., Ircing, P., Khudanpur, S.: Hidden markov models for automatic annotation and contentbased retrieval of images and video. In: *ACM Conference on Special Interest Group on Information Retrieval (SIGIR)*, Brazil (2005)
11. Chapelle, O., Haffner, P., Vapnik, V.: Support vector machines for histogram-based image classification. *IEEE Transactions on Neural Networks* 10, 1055–1064 (1999)
12. Gao, Y., Fan, J., Xue, X., Jain, R.: Automatic image annotation by incorporating feature hierarchy and boosting to scale up svm classifiers. In: *Proceedings of the 14th Annual ACM International Conference on Multimedia, MULTIMEDIA 2006*, New York, NY, USA, pp. 901–910 (2006)
13. Qi, X., Han, Y.: Incorporating multiple svms for automatic image annotation. *Pattern Recognition* 40, 728–741 (2007)
14. Tsai, C.: Stacked generalization: A novel solution to bridge the semantic gap for contentbased image retrieval. *Online Inf. Rev.* 27, 442–445 (2003)
15. Yu, H., Wolf, W.: Scenic classification methods for image and video database. In: *Proceedings of the SPIE International Conference on Digital Image Storage and Archiving Systems*, pp. 363–371 (1995)
16. Li, J., Wang, J.: Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25, 1075–1088 (2003)
17. Modestino, J., Zhang, J.: A markov random field model-based approach to image interpretation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14, 606–615 (1992)
18. Zhao, Y., Zhao, Y., Zhu, Z.: Tsvm-hmm: Transductive svm based hidden markov model for automatic image annotation. *Expert Systems with Applications* 36, 9813–9818 (2009)
19. Feng, S., Manmatha, R., Lavrenko, V.: Multiple bernoulli relevance models for image and video annotation. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 1002–1009 (2004)

Video Deblurring and Super-Resolution Technique for Multiple Moving Objects

Takuma Yamaguchi¹, Hisato Fukuda², Ryo Furukawa³,
Hiroshi Kawasaki⁴, and Peter Sturm⁵

¹ Research & Development Center, The Nippon Signal Co., LTD
1836-1 Oaza Ezura, Kuki, Saitama 346-8524 Japan
ymgc-tkm@signal.co.jp

² Saitama University, Department of Information and Computer Science
255 Shimo-okubo, Sakura-ku, Saitama, 338-8570, Japan
fukuda@cv.ics.saitama-u.ac.jp

³ Faculty of Information Sciences, Hiroshima City University
3-4-1 Ozuka-higasi, Asaminami-ku, Hiroshima, 731-3194 Japan
ryo-f@hiroshima-cu.ac.jp

⁴ Faculty of Engineering, Kagoshima University
1-21-24 Korimoto, Kagoshima City, Kagoshima, 890-0065 Japan
kawasaki@ibe.kagoshima-u.ac.jp

⁵ INRIA Grenoble – Rhone-Alpes
655 Avenue de l'Europe, 38330 Montbonnot St Martin, France
peter.sturm@inrialpes.fr

Abstract. Video camera is now commonly used and demand of capturing a single frame from video sequence is increasing. Since resolution of video camera is usually lower than digital camera and video data usually contains a many motion blur in the sequence, simple frame capture can produce only low quality image; image restoration technique is inevitably required. In this paper, we propose a method to restore a sharp and high-resolution image from a video sequence by motion deblur for each frame followed by super-resolution technique. Since the frame-rate of the video camera is high and variance of feature appearance in successive frames and motion of feature points are usually small, we can still estimate scene geometries from video data with blur. Therefore, by using such geometric information, we first apply motion deblur for each frame, and then, super-resolve the images from the deblurred image set. For better result, we also propose an adaptive super-resolution technique considering different defocus blur effects dependent on depth. Experimental results are shown to prove the strength of our method.

1 Introduction

Demand for retrieving a high quality single image from video sequence is increasing, such as surveillance and handheld video capture and so on. Since image quality of video camera is usually lower than digital camera, simple frame capture is often insufficient for actual purpose. Although the main reason of the low quality

of video data is a low resolution of video camera, motion blur is another important reason of degradation; it commonly occurs because video usually captures moving object, whereas, still camera mainly captures static scene only. Another problem on quality of video data is narrow depth of field; it is also common because video camera requires high frame-rate with fast shutter speed, resulting in wide aperture. Because of the narrow depth of field, the scene other than target object is blurred by defocus blur. Thus, simple frame capture can produce only low quality image and image restoration technique is inevitably required.

To deal with the problem mentioned above, hybrid camera systems are proposed [1,2]. However, since those systems require additional sensors, the systems become complicated and the technique cannot be applied for common video data. On the other hand super-resolution technique using several input frames are proposed. However, most of them does not consider motion blur and only several papers take the problem into account; they treat motion blur as noise [3]. Therefore, quality of image restoration is limited.

In this paper, we propose a method to restore a sharp and high-resolution image from a video sequence by applying a motion deblurring technique for each frame followed by super-resolution technique for multiple frames. To conduct a motion deblur from an image, motion information is required. Since typical device of deblurring techniques is a still camera, they assume long exposure time and complicated camera motion; thus, sophisticated blind kernel estimation technique is usually required. To the contrary, with video camera, motion is usually small and simple for each frame. One important problem for video is that several objects move independently. In our method, by taking account of such feature of video camera, we propose a motion deblurring technique using optical flow of the scene with scene segmentation technique.

In terms of super-resolution of the image sequence, sub-pixel registration is required and it is usually difficult to achieve with blurry image. Since motion blur is reduced by our method in the first step, the problem is greatly reduced. In addition, since the scene contains several independently moving objects, segmentation and area based registration for each segment is required; it is efficiently solved by our pixel-based plane approximation technique. Further, image quality is further improved by considering the different defocus blur for each segment dependent on different depth with our adaptive super-resolution technique.

2 Related Work

In terms of deblurring techniques for motion blur, since the blur is a convolution process, restoration technique has been proposed as a deconvolution technique for known kernel [4,5]. If the kernel is unknown, such condition is common for usual photos, the problem is ill-conditioned and it cannot be solved without additional information [6]. For simple and straight-forward solution, an additional sensor is used to estimate the blur kernel [1,7]. Recently, blind deconvolution techniques using the information of natural scene, i.e., “heavy tailed distribution in the gradients” are proposed [8,9,10,11]. We also use the same knowledge to estimate the motion blur kernel.

Generally, the main reason of motion blur is assumed to be a camera motion, such as camera shake, thus, previous technique usually uses a single blur kernel for deblurring. Currently several researches are proposed considering object motion in the scene [7,12]. In addition, more general cases, such as an independent blur kernel for each depth of an object is proposed [13]. We also estimate independent blur kernel for each segment.

In terms of super-resolution techniques, reconstructing a high-resolution image from multiple low-resolution images is intensively researched [14,15,16]. In those techniques, it is assumed that scenes are either static or dynamic, but consist of single depth or planar objects with little motion, and the camera is also assumed to be static. With such assumptions, registration between frames can be simplified and it can be done with sufficient accuracies with 2D affine or homography transformation. However, for applying techniques to more general purposes, it is necessary to allow 3D scenes containing multiple independently moving objects, non-rigid motion objects (*e.g.* cloths), etc. With existing super-resolution techniques, it is difficult to achieve this, because of significant appearance changes caused by objects' motion and viewpoint changes. To perform super-resolution for such objects or scenes, 3D information should be considered. Tung et al. [17] have applied super-resolution technique to construct a high-resolution 3D video. However, the technique is based on approximating 3D objects by triangular patches, and thus, accurate and dense 3D data is required; it cannot be easily acquired in general.

The technique to achieve both motion deblur and super-resolution is also proposed by Tai et al. [2]. The central idea is similar to ours, however, the method to estimate the motion of the scene is totally different; we estimate it only from video data, whereas Tai et al. use additional device as hybrid system.

3 Algorithm Overview

A simple solution to restore the images that are degraded by blur kernels per each frame and object is to prepare each kernel for calculation. However, the considered input is a video sequence captured by a handheld camera, and thus, such blur kernels are not usually given. In this paper, since the input is a video sequence, we estimate those blur kernels for each segmented region of objects in the scene; those regions are detected by segmentation using optical flow field.

In terms of motion deblurring, we assume that the blur of the region to be combination of motion blur and defocus blur, where defocus blur is constant for each region. With such video data, feature points are also blurry because of motion blur and it is difficult to achieve high accuracy to detect them, however, optical flow field can be accurately acquired with area based method. Therefore, we use the optical flow field to estimate motion blur kernel.

On the other hand, restoration of low resolution image with defocus blur has been researched for long time, typically via super-resolution techniques; it is known that the quality is low if only a single image is used, and thus, many techniques using multiple images and MAP estimation are proposed to achieve reasonable results [15,16]. To super-resolve images from low-resolution and blurred

images, sub-pixel registration is required. In previous methods, where the scene is assumed to be a single plane, accurate registration can be easily achieved. However, natural scenes consist of multiple dynamic 3D objects, and thus, achieving an accurate and robust registration is not easy. In this paper, we propose a plane based registration method to achieve sub-pixel accuracy for registration of all the pixels in the images.

As already described, we assume that image blur to be combination of motion and defocus blur. Since we assume that there are several objects at different depths in the scene, all objects do not suffer from the same defocus blur. Therefore, we propose an *adaptive deblurring method* to change kernels for defocus blur adaptively for each object. Certainly, estimating blur kernels for each object is not easy, therefore, for simplicity, we assume in this paper that the kernels of defocus blur can be described as one-parameter point spread functions (Bessel function). Since we consider moving objects in the scene, defocus blur kernels may vary for each frame. However, since we use between 20 and 40 frames for super-resolution, i.e., just 1 to 2 seconds of video, we assume that large changes of defocus blur kernels are unlikely, and thus, we use the same kernel for the process. Actual algorithm is as follows.

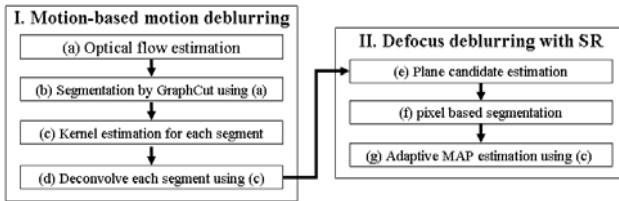


Fig. 1. Flow of the deblurring process

First, we estimate optical flow field of the input image sequence by using block matching technique. (Fig. [II\(a\)](#)). The optical flow field is segmented by graph cut method, where each of the regions include almost constant flow vectors (Fig. [II\(b\)](#)). Then, an initial blur kernel is estimated for each segments (Fig. [II\(c\)](#)). Simple super-resolution may not bring good results when the input images contain motion blur, because it is difficult to achieve high accuracy registration with such blurry images. Therefore, motion deblurring technique is applied before a super-resolution (Fig. [II\(d\)](#)). Finally, these frame-wise deblurred results are further improved by using super-resolution technique, simultaneously improving the resolution and defocus blur (Fig. [II](#)).

4 Motion Deblurring for Multiple Moving Objects

In this paper, we estimate the motion of each region by segmenting the optical flow field, and use the flow vectors for the regions to estimate motion blur kernels. For simplicity, we model image blur as a convolution of a line-shaped motion blur kernel and one-parameter isotropic defocus blur kernel. Certainly, a line-shaped

motion blur kernel sometimes results in an insufficient quality, especially for a large camera motion (e.g., severe ringing effects), however, camera motion is usually small and simple in our research, because all images are captured by video camera where a shutter speed is usually faster than 1/60 to keep 30 fps, and such simple kernel can achieve enough restoration in reality.

As the line-shaped motion blur kernel estimation, we use a direction of optical flow for its direction, and the knowledge of the derivatives histogram of natural scene to estimate the scaling parameter; note that such scaling parameter estimation is currently common and used by several research groups [10, 18]. The actual kernel estimation proceeds as follows:

1. The optical flow field is estimated for all the images based on block matching.
2. Input images are segmented into regions, each of which has almost constant motion vectors.
3. For each region, a line-shaped motion blur kernel is estimated from optical flow and the derivatives histogram of the image.
4. Motion blur is reduced by deconvolution algorithm by using the line-shaped blur kernels.

These processes are explained in the following sections in detail.

4.1 Segmentation of Blurry Image Sequence

An input data of the proposed method is a captured sequence of images. The image may be captured by a static or moving camera. The captured scene may include multiple objects that may be static or moving. Therefore, segmentation for each object is required. Since input image is blurry, feature based method may not work, and thus, area based approach is used. In this paper, optical flow field is obtained by pyramid based block matching method. Then, multi-value graph-cut method is applied to those flow field. In our implementation, we put a large value on a direction rather than a length of the optical flow for data-term of graph-cut from our experience of several experiments. We also assume only 3 to 5 segments in the scene for fast calculation.

4.2 Blur Kernel Estimation Using Optical Flow

For each extracted region, blur kernel is estimated. In our research, we assume that the shape of the motion blur kernel to be linear as mentioned above. We use a direction of optical flow for its direction, and estimate the scaling parameter by using the knowledge of the derivatives histogram of natural scene; i.e., the derivatives histograms of the scene for all directions are usually the same in natural scene. Therefore, actual algorithm is as follows.

First, we calculate the derivatives histogram along optical flow vector direction. Then, we add blur to the perpendicular direction by changing the kernel size so that the both derivative histograms become similar. Fig 2(a) and (b) show the both derivatives histogram along optical flow direction and its perpendicular direction. Fig 2(c) shows the derivatives histogram along the perpendicular direction after applying the estimated blur kernel. We can clearly see that the shapes of Fig 2(a) and (c) look similar.

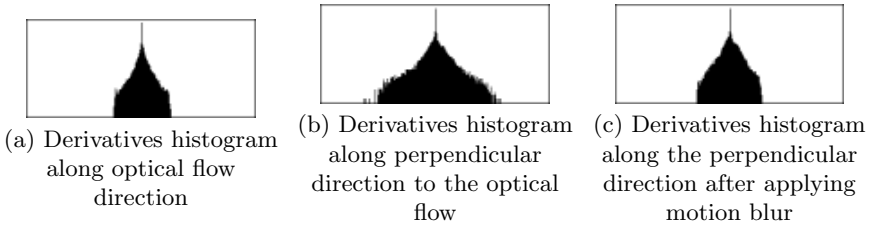


Fig. 2. Motion blur kernel estimation

4.3 Motion Deblurring for Each Segment

In terms of deconvolution algorithm, several techniques exist; Iterative Back Projection [5] is applied in our approach.

5 Super-Resolution Technique for Multiple Depth

We adopt a multi-frame super-resolution technique to restore both low-resolution and defocus blur. To realize an efficient removal of defocus blur, we first carry out a piecewise planar segmentation of the scene, in order to accomplish accurate registration and set appropriate blur kernels dependent on depth in 3D scenes. The segmentation algorithm basically consists of two steps; (1) plane candidate generation by using feature tracking results and (2) pixel-based segmentation by minimizing re-projection errors. For super-resolution, we use a MAP image reconstruction formulation with the registration result for each segment.

5.1 Estimating Candidate Planes Based on Feature Point Tracking

A number of studies have already been reported related to the extraction of planes from the scene for the purpose of 3D reconstruction [19, 20, 21]. In these studies, planar areas are extracted as patches by clustering feature points. However, in practice, it is often difficult to perform an accurate plane-based approximation because individual feature point tracks are easily affected by outliers, the aperture problem and view-dependent appearance changes, even if the global ego-motion estimation is accurate. In addition, since features are often not detected along object boundaries, patch creation is another difficult problem.

In this paper, we propose a pixel-based plane estimation which is more suitable than a patch-based technique. More specifically, instead of dividing the scene into

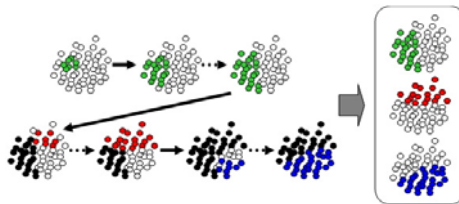


Fig. 3. Candidate plane detection

patches, candidate planes are first extracted, each of which is defined by a group of tracked feature points included in a single plane. To achieve good results, a sufficient number of candidate planes should be extracted to approximate the 3D scene. A simple solution is to extract as many planes as possible from all combinations of the feature points. On the other hand, the smaller the number of candidate planes, the more efficient the computation. Therefore, we propose an efficient method to reduce the number of candidate planes to approximate the 3D scene by using the knowledge that neighboring feature points usually belong to the same plane.

Our candidate plane estimation method is described in Algorithm 1. First, corresponding feature points between input frames are computed. Then, an initial candidate plane which described by feature point tracks is generated. Using the tracks, the homography matrices between the base frame and the other frames are calculated. Next, the candidate plane is updated. Feature points whose evaluation values are less than the threshold value (0.2 pixel in our case), are added to the plane. We use the average of the re-projection errors of all the corresponding points as the evaluation value. And then, the homography matrix calculation and updating the candidate plane are iterated until the feature point tracks on the plane are converged. Repeating this manner, candidate planes describing the scene are obtained. Fig. 3 shows an example for the generation of three groups, where the black points represent the feature points which are already calculated or assigned to some planes, and the white points represent unselected and unlabeled points.

5.2 Pixel-Based Segmentation by Minimization of Re-projection Errors

Since the candidate planes (groups of feature points each of which is included in a single plane) extracted by the aforementioned method are represented as groups of feature points rather than explicit patches, the dense pixel correspondence is not yet determined at this stage. Since transformation parameters of each candidate plane between frames are calculated in the previous step, pixel-based correspondences can be estimated by assigning each pixel to one of the candidate planes by minimizing the re-projection error using the parameters.

In this paper, the homography matrices obtained from the candidate planes are used as transformation parameters. Then, the differences of intensity for each pixel from a reference frame to all other frames are computed, the average of the differences is stored for each plane, and the pixel is assigned to the plane

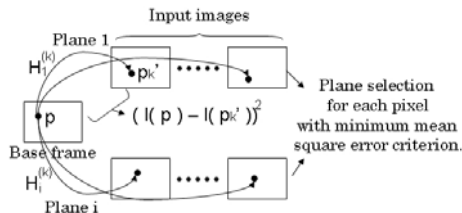


Fig. 4. Plane selection for each pixel

for which that average is the smallest. The actual calculation is as follows. We denote the number of input frames as N , the homography matrix (as obtained from the i -th candidate plane, *i.e.*, the i -th group of feature points) between the reference frame and the k -th frame as $\mathcal{H}_i^{(k)}$, and the respective intensity levels of arbitrary points in the reference frame and the k -th frame as $I(\cdot)$ and $I^{(k)}(\cdot)$, respectively. Then the following equation is obtained for each pixel in the reference frame.

$$\hat{i}\mathbf{p} = \underset{i}{\operatorname{argmin}} \left[\frac{\sum_{k=1}^M \left\{ I(\mathbf{p}) - I^{(k)}(\mathcal{H}_i^{(k)} \mathbf{p}) \right\}^2}{M} \right] \quad (1)$$

Here, $M(\leq N)$ denotes the number of frames for which the pixels were effective before the projection (in other words, the pixels were within the image), and \mathbf{p} represents a coordinate vector. By finding the minimum projection difference, each pixel is assigned to plane $\hat{i}\mathbf{p}$. Note that since we can reject pixels whose difference measure is large, our method can handle occlusions. The process is shown in Fig. 4.

5.3 Adaptive SR by MAP Estimation

We use a maximum a posteriori (MAP) image reconstruction formulation for multi-frame super-resolution as follows:

$$\hat{X} = \underset{X}{\operatorname{argmin}} \left[\sum_{k=1}^N \|D_k H_k F_k X - Y_k\|_2^2 + \lambda \|I X\|_2^2 \right] \quad (2)$$

Algorithm 1. Candidate plane estimation.

- 1: X is defined as the set of all corresponding feature point tracks across input frames.
 - 2: $P(x)$ is defined as a predicate that is true if point track x is not selected and unlabeled.
 - 3: **while** $\exists x \in X; P(x)$ **do**
 - 4: Select a feature point track $a(\subseteq \{x \in X; P(x)\})$ and the k nearest neighbors $b(\subseteq X)$ (in this paper $k := 7$).
 - 5: $A^{(0)} := \phi$, $A^{(1)} := a \cup b$, $i := 1$
 - 6: **while** $A^{(i)} \neq A^{(i-1)}$ **do**
 - 7: Compute the homography matrix \mathcal{H} of $A^{(i)}$ for each frame.
 - 8: $A^{(i+1)} := \phi$
 - 9: **for** $\forall y \in X$ **do**
 - 10: **if** Adequateness of \mathcal{H} for $y \geq \text{threshold}$ **then**
 - 11: $A^{(i+1)} := A^{(i+1)} \cup y$
 - 12: **end if**
 - 13: **end for**
 - 14: $i := i + 1$
 - 15: **end while**
 - 16: $A^{(i)}$ is a group of feature point tracks residing in the same plane.
 - 17: **end while**
-

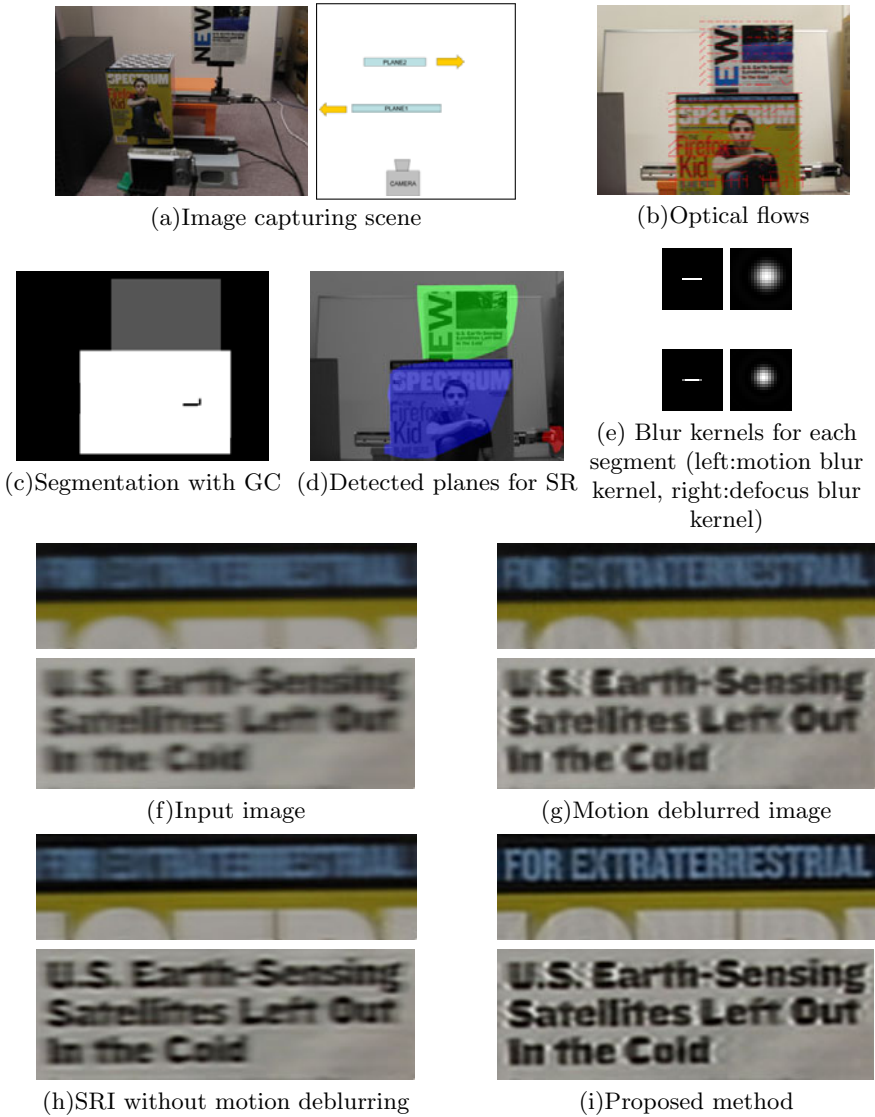


Fig. 5. Multiple object motion by motorized stage

where F_k is the geometric motion operator between the high-resolution (HR) frame X and the k th low-resolution (LR) frame Y_k , H_k is the defocus blur matrix representing the camera's point spread function and D_k stands for the decimation matrix (F_k is previously estimated, see Sec. 5.2). $\|\Gamma X\|^2$ is the Tikhonov regularization cost function and λ is the regularization parameter. Generally, a high-pass operator is used as Γ ; we use the Laplacian.

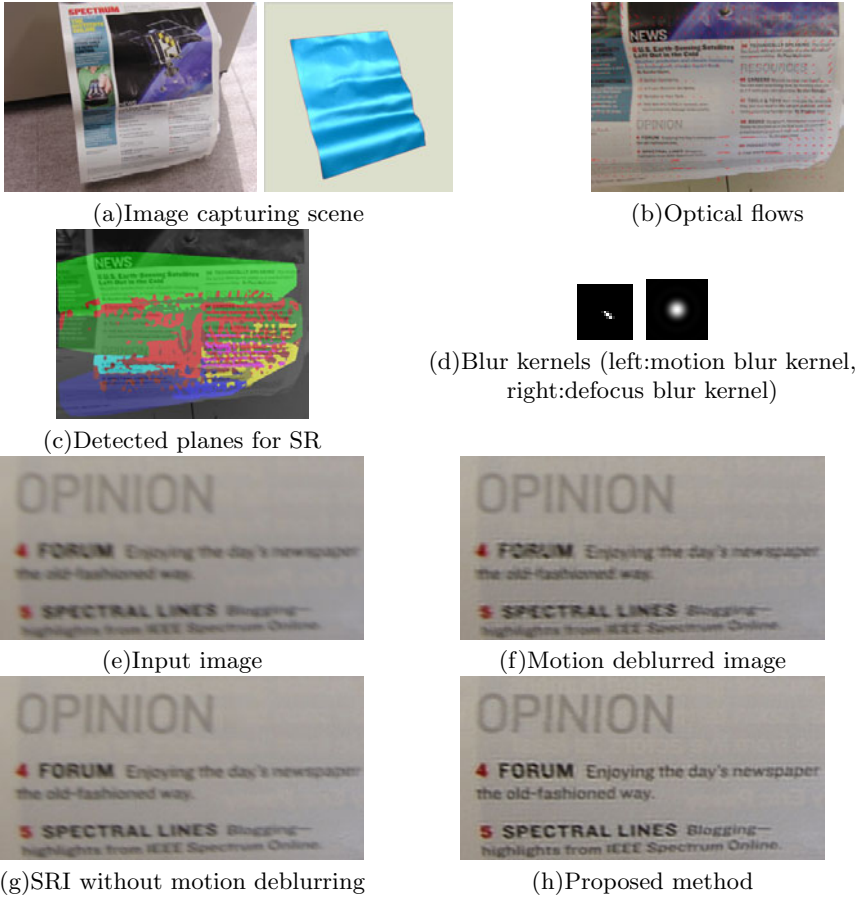


Fig. 6. Motion deblur and super-resolution for curved surface

If we assume that all the decimation operations are the same (i.e. $\forall k, D_k = D$) and all the blur operations are the same (i.e. $\forall k, H_k = H$), (2) may be written as

$$\hat{X} = \operatorname{argmin}_X \left[\sum_{k=1}^N \|DF_k HX - Y_k\|_2^2 + \lambda \|GX\|_2^2 \right]. \quad (3)$$

We decompose this minimization problem into the following two separate steps, as suggested in [3].

1. Compute a defocus blurred HR image $\hat{Z}(= H\hat{X})$ from the LR images.
2. Estimate the HR image \hat{X} from the defocus blurred HR image \hat{Z} .

In this paper \hat{Z} is calculated by solving the following minimization problem:

$$\hat{Z} = \operatorname{argmin}_Z \left[\sum_{k=1}^N \|DF_k \hat{Z} - Y_k\|_2^2 \right]. \quad (4)$$

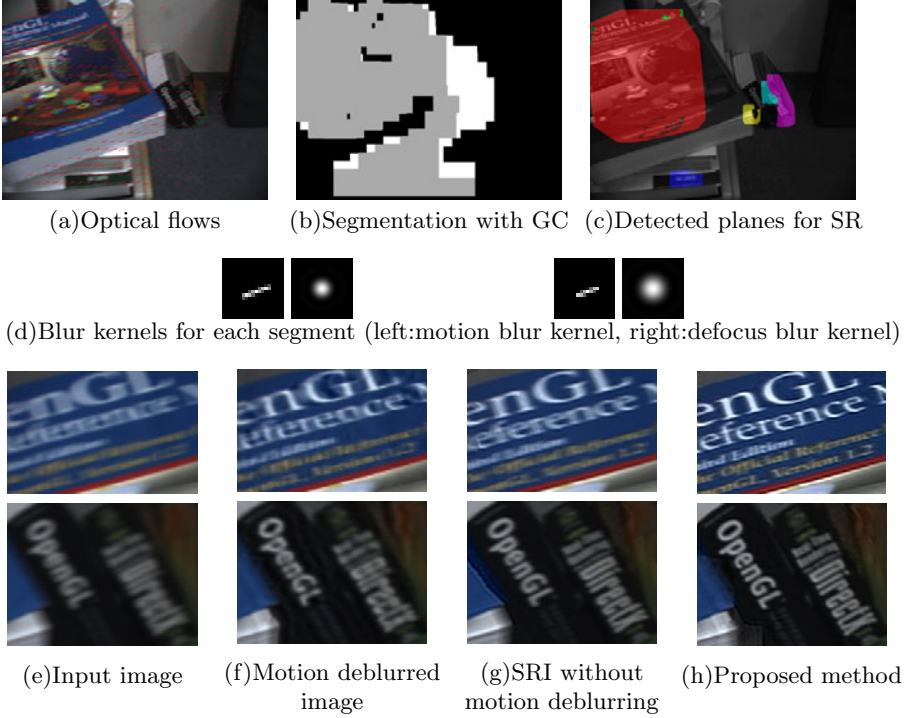


Fig. 7. Video data capture by handheld camera

In the deblurring step, the deblurred HR image \hat{X} is obtained through the following formulation:

$$\hat{X} = \underset{X}{\operatorname{argmin}} \left[\|W(HX - Z)\|_2^2 + \lambda \|GX\|_2^2 \right]. \quad (5)$$

where W is a diagonal matrix, each of whose diagonal values equals the number of measurements for one pixel. With this formulation, different blur kernels can be set to each pixel.

6 Experiments

6.1 Evaluation of the Method Using Real-Data

To test the effectiveness of the method, we conducted experiments using motorized stage. In this data, the scene consists of two planes with texture as shown in Fig. 5(a). We set the nearest plane to be in focus and the other plane undergo a depth-dependent defocus blur by the camera aperture. We moved the two objects with different speed and different direction by two different motorized stages. The super-resolution image (SRI) with our method is shown in Fig. 5(i).



Fig. 8. Video data captured by static camera

We can still observe small ringing effects remaining near edges, however, strong motion blur is removed and super-resolution is successfully conducted.

Next, we apply the technique to curved surfaces. The result is shown in Fig. 6. In Fig. 6(c), we can see that the scene is successfully segmented into several planes to approximate the curved surfaces. In Fig. 6(h), we can clearly see that the motion blur is removed and super-resolution is successfully applied even if the shape has no planer area.

6.2 Handheld Video Data Scene

In this experiment, we conducted an experiment using a handheld video camera as shown in Fig. 7(a). The motion deblurred image with our method is shown

in Fig. 7(f). We can see that motion blur was successfully removed. The result of plane segmentation applied on the motion deblurred image sequence and the final super-resolved image are shown in Fig. 7(c) and (h). Even for such natural sequence captured by handheld video, each plane was successfully segmented and super-resolution is successfully achieved.

The super-resolution image without motion deblurring is shown in Fig. 7(g). We can clearly see that our method gives the best restoration.

6.3 Multiple Moving Objects Captured by Static Camera

Finally, we conducted the same experiment with static camera and multiple moving objects. Fig. 8(a) shows example and optical flows of the input data. Fig. 8(f) shows a motion deblurred image and Fig. 8(h) shows the final result by applying adaptive MAP estimation on the motion deblurred images. Fig. 8(g) shows the result of simple super-resolution and we can confirm that our method achieved the best restoration.

7 Conclusion

In this paper, we propose a method to restore a sharp and high-resolution image from video data captured by a handheld camera in which both independent motion and defocus blur are observed. The method is based on a motion deblurring technique using estimated blur kernels for each frame and object and super-resolution technique with adaptive defocus blur kernel. A motion blur kernel is efficiently estimated by using optical-flow and natural scene statistics and motion blur is reduced by a deconvolution algorithm. A defocus blur is removed by an adaptive MAP estimation technique with pixel-wise plane segmentation method. We conducted several experiments using real data which successfully show the effectiveness of our method. Extended research on deforming object with independent motion blur is our next step.

Acknowledgment. This work was supported in part by SCOPE No.101710002 and Grant-in-Aid for Scientific Research No.072103013, No.19700098 and 21700183 in Japan.

References

1. Ben-Ezra, M., Nayar, S.: Motion Deblurring using Hybrid Imaging. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. I, pp. 657–664 (2003)
2. Tai, Y.W., Du, H., Brown, M.S., Lin, S.: Image/video deblurring using a hybrid camera. In: CVPR (2008)
3. Farsiu, S., Robinson, M.D., Elad, M., Milanfar, P.: Fast and robust multiframe super resolution. *IEEE Trans. on Image Processing* 13, 1327–1344 (2004)
4. Lucy, L.: An iterative technique for the rectification of observed distributions. *J. of Astronomy* 79, 745–754 (1974)

5. Irani, M.I., Peleg, S.: Improving resolution by image registration. *CVGIP* 53, 231–239 (1991)
6. Levin, A., Weiss, Y., Durand, F., Freeman, W.: Understanding and evaluating blind deconvolution algorithms. In: *CVPR*, pp. 1964–1971 (2009)
7. Li, F., Yu, J., Chai, J.: A hybrid camera for motion deblurring and depth map super-resolution. In: *CVPR* (2008)
8. Sroubek, F., Gabriel, C., Flusser, J.: A unified approach to superresolution and multichannel blind deconvolution. *IEEE Trans. on Image Processing* 16, 2322–2332 (2007)
9. Fergus, R., Singh, B., Hertzmann, A., Roweis, S.T., Freeman, W.T.: Removing camera shake from a single photograph. In: *SIGGRAPH*, pp. 787–794 (2006)
10. Shan, Q., Jia, J., Agarwala, A.: High-quality motion deblurring from a single image. In: *ACM Transactions on Graphics, SIGGRAPH* (2008)
11. Levin, A., Fergus, R., Durand, F., Freeman, W.: Image and depth from a conventional camera with a coded aperture. In: *SIGGRAPH* (2007)
12. Levin, A., Sand, P., Cho, T.S., Durand, F., Freeman, W.T.: Motion-invariant photography. In: *SIGGRAPH*, pp. 1–9 (2008)
13. Cho, S., Matsushita, Y., Lee, S.: Removing non-uniform motion blur from images. In: *ICCV*, pp. 1–8 (2007)
14. Farsiu, S., Elad, M., Milanfar, P.: A practical approach to superresolution. In: *Visual Communications and Image Processing*, vol. 6077 (2006)
15. Katsaggelos, A.K., Molina, R., Mateos, J.: *Super Resolution of Images and Video*. Morgan & Claypool Publishers, San Francisco (2006)
16. Park, S.C., Park, M.K., Kang, M.G.: Super-resolution image reconstruction: a technical overview. *IEEE Signal Processing Magazine* 20, 21–36 (2003)
17. Tung, T., Nobuhara, S., Matsuyama, T.: Simultaneous super-resolution and 3d video using graph-cuts. In: *CVPR* (2008)
18. Levin, A.: Blind motion deblurring using image statistics. In: *NIPS* (2006)
19. Zucchelli, M., Santos-Victor, J., Christensen, H.: Multiple plane segmentation using optical flow. In: *BMVC*, pp. 313–322 (2002)
20. Dick, A., Torr, P., Cipolla, R.: Automatic 3d modelling of architecture. In: *BMVC*, pp. 372–381 (2000)
21. Bartoli, A.: Piecewise planar segmentation for automatic scene modeling. In: *CVPR*, pp. 283–289 (2001)

Sparse Source Separation of Non-instantaneous Spatially Varying Single Path Mixtures

Albert Achtenberg and Yehoshua Y. Zeevi

Department of Electrical Engineering, Technion, 3200 Haifa, Israel

Abstract. We present a method for recovering source images from their non-instantaneous single path mixtures using sparse component analysis (SCA). Non-instantaneous single path mixtures refer to mixtures generated by a mixing system that spatially distorts the source images (non-instantaneous and spatially varying) without any reverberations (single path/anechoic). For example, such mixtures can be found when imaging through a semi-reflective convex medium or in various movie fade effects. Recent studies have used SCA to separately address the time/position varying and the non-instantaneous scenarios. The present study is devoted to the unified scenario. Given n anechoic mixtures (without multiple reflections) of m source images, we recover the images up to a limited number of unknown parameters. This is accomplished by means of correspondence that we establish between the sparse representation of the input mixtures. Analyzing these correspondences allows us to recover models of both spatial distortion and attenuation. We implement a staged method for recovering the spatial distortion and attenuation, in order to reduce parametric model complexity by making use of descriptor invariants and model separability. Once the models have been recovered, well known BSS tools and techniques are used in recovering the sources.

1 Introduction

In many real world applications, input signals to a system are mixtures of some more meaningful source signals (image layers, different speakers, musical instruments, etc). The problem of recovering m sources from n mixtures with only limited knowledge of the mixing process is well known as the Blind Source Separation (BSS) problem. Most research in the field of BSS has focused on instantaneous and time invariant cases [13]. Convolutive mixtures are being currently extensively studied, using SCA as well as the popular independent component analysis (ICA) techniques [16]. However, only few recent studies addressed the general¹ time varying scenario [6, 7, 18] or the general non-instantaneous case [8, 9]. Some specific parametric families of the single path problem were also addressed by recent studies [4, 5]. The term *single path* (or anechoic) mixtures

¹ By 'general' we mean that the spatially dependent model is not limited to the linear or convolutive scenarios but may be arbitrarily non-linear.

is used throughout this paper to address the general case that covers both non-instantaneous and time varying anechoic mixtures. Single path time/position varying mixtures are common physical phenomena. One such example is of images taken through a semi reflective distorting medium (lens), where the target and superimposed reflected image are differently affected [9]. Another example is audio signals distorted by the Doppler effect and attenuated by the scattering medium (assuming an anechoic medium).

In this paper we propose a framework which copes with the general single path scenario. The proposed method assumes that the mixtures and the temporal distortions are known up to a small number of parameters. However, it can be extended to nonparametric models as well. The following preliminary remarks are in place:

- The proposed method is applicable to the general problem of m sources and n mixtures. To simplify the presentation we address the case of 2 mixtures and 2 sources
- We consider two-dimensional signals (i.e. images). One-dimensional signals (such as audio signals) are addressed in [3.1]

The rest of the paper is organized as follows. In Section 2 we define the problem at hand. In Section 3 we present the sparsification and alignment technique which is the key to the separation procedure. In Section 4 we outline the separation process. Representative results are presented in Section 5.

2 Problem Definition

The single path BSS problem can be formulated as follows [9]:

$$\begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} g_{11} & \cdots & g_{1m} \\ \vdots & \ddots & \vdots \\ g_{n1} & \cdots & g_{nm} \end{bmatrix} \star \begin{bmatrix} s_1 \\ \vdots \\ s_m \end{bmatrix} \quad (1)$$

where $\{z_i\}$ are the given mixtures, $\{s_i\}$ are the unknown sources and $\{g_{ij}\}$ are the unknown mixing kernels. The mixing operator \star is considered to be an integral operation with multidimensional kernel function, possibly non-linear. For the special case of single path mixtures, the kernel functions $\{g_{ij}\}$ are of the special form:

$$g_{ij}(x, y, u, v) = \hat{a}_{ij}(u, v)\delta(u - T_{ij}^x(x, y), v - T_{ij}^y(x, y)) \quad (2)$$

Applying the kernel to any of the source signals produces the anticipated result:

$$\begin{aligned} (g_{ij} \star s_i)(x, y) &= \int \int s(u, v)\hat{a}_{ij}(u, v)\delta(u - T_{ij}^x(x, y), v - T_{ij}^y(x, y))dudv \\ &= s(T_{ij}^x(x, y), T_{ij}^y(x, y))\hat{a}_{ij}(T_{ij}^x(x, y), T_{ij}^y(x, y)) \\ &= a_{ij}(x, y)s(T_{ij}^x(x, y), T_{ij}^y(x, y)) = w_{ij}(x, y) \end{aligned} \quad (3)$$

$$a_{ij}(x, y) = \hat{a}_{ij}(T_{ij}^x(x, y), T_{ij}^y(x, y)) \quad (4)$$

where \hat{a}_{ij} and a_{ij} are the attenuation and the spatially distorted attenuation, respectively, and T_{ij} is the spatial distortion.

By analyzing this result we notice that every single mixed signal component w_{ij} is an attenuated and spatially distorted version of the source s_i . The attenuation factor is a_{ij} , while the spatial distortion is $(x - T_{ij}^x(x, y), y - T_{ij}^y(x, y))$. The mixed signals can be written as:

$$z_i(x, y) = \sum_j w_{ij}(x, y) = \sum_j a_{ij}(x, y) s_j(T_{ij}^x(x, y), T_{ij}^y(x, y)) \quad (5)$$

The following sections deal with the problem of estimating mixing parameters a_{ij} and T_{ij} , and consequently estimating the source signals s_i .

3 Mixture Alignment

As in every mixed signal, the sources may have undergone different spatial distortion. We have no trivial way to track the contribution of a sample originated in one source signal across the different mixed signals. In fact, this prohibits us from directly applying known SCA BSS techniques. To overcome this limitation, we apply a signal alignment scheme proposed by [8,9] and extend it to the general single path scenario. This approach uses local features to find correspondences across the mixed signals' sparse representations. We denote the correspondences between mixed signals z_i and z_j as the set of sample index pairs $\left\{ (p_k^{ij}, q_k^{ij}) \right\}_1^K$. The correspondence process requires the sparsifying function to:

1. Retain significant signal details
2. Repeat itself across mixtures
3. Be easy to track across mixtures.

These requirements have to be satisfied in the presence of varying attenuation and spatial distortion. When dealing with images, one candidate that yields such representation is the Scale Invariant Feature Transform (SIFT) [12]. SIFT key-point detection can be regarded as a sparsification of the source, while the SIFT descriptor and key-point matching scheme assume the roles of the local feature detector and feature corresponder. Although SIFT can be replaced by any other detector/descriptor pair, it provides scale, translation, rotation and amplitude invariance which are key features for the success of matching distorted signals. Finding a significant number of key-point matches enables us to proceed to the separation stage. An alternative related candidate is the wavelet-type corner detector [2].

Having established the image correspondences, we can track source originated features from one mixture to another and thus apply the sparse separation approach [2]. This approach states that a sparse representation of the mixed signal will locally resemble only one of the original sources. Let \mathfrak{F} be a sparsifying function and $\{c_k^i\}$ a set of points, where the sparse representation of a signal is non

zero. The following equation should hold for a sufficiently large number of points in the sparse representation:

$$\exists l : \mathfrak{F} \{z_i(\cdot)\} (c_k^i) \approx \mathfrak{F} \{w_{il}(\cdot)\} (c_k^i) \quad (6)$$

By applying this logic to the mixed signals' correspondences we may conclude that both correspondence samples are originated in the same source signal, s_l :

$$\begin{aligned} \exists l : \mathfrak{F} \{z_i(\cdot)\} (p_k^{ij}) &\approx \mathfrak{F} \{w_{il}(\cdot)\} (p_k^{ij}) \\ \mathfrak{F} \{z_j(\cdot)\} (q_k^{ij}) &\approx \mathfrak{F} \{w_{jl}(\cdot)\} (q_k^{ij}) \end{aligned} \quad (7)$$

3.1 1D/N-D Signals

The same approach can be applied to one-dimensional, or to N-dimensional signals. However, some modification have to be made to fit the nature of audio signals:

1. Sparsification - Extensive research has been conducted in the field of audio feature detection [110]. Existing results can be used as a good sparsification/key-frame detection algorithms.
2. Descriptor - While the SIFT descriptor for images relies on local gradient histograms, local audio frame descriptor should rely on the spectral contents of the frame. This is due to the oscillatory "textural" nature of audio signals. Tolerance to local temporal distortion (Doppler effect) can be achieved by spectrally normalizing the descriptor.
3. Matching - If we assume that temporal distortion does not change the temporal order of audio events, we can use techniques such as dynamic time warp (DTW) [17] or spectral matching [11] to improve the matches by imposing constrains on the matching process.

4 Separation Process

Having the mixtures $\{z_i\}$ locally aligned using key-point correspondences, the separation can be dealt with by using the staged approach [9]:

1. Estimate the mixing system ($\{a_{ij}(\cdot)\}, \{T_{ij}^x(\cdot), T_{ij}^y(\cdot)\}$)
2. Invert the mixing operation to recover the sources.

4.1 Assumptions

1. Both spatial distortion, T , and attenuation, a , factors can be represented by means of parametric representations.
2. Images can be sparsely represented using some kind of sparsification transformation
3. Each of the source images dominates at least some local segments of the mixtures
4. Sources have undergone significantly different spatial distortions.

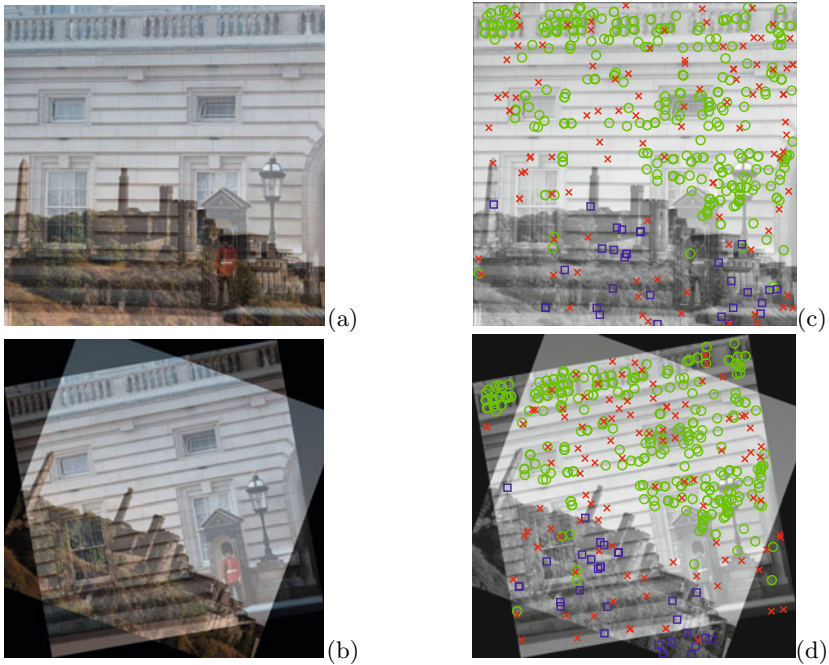


Fig. 1. An example of data separation. (a)-(b) Two mixtures of the sources from Fig. 2 (c)-(d) Classified key-points on both mixtures. Green circles are classified as belonging to source #1. Blue squares are classified as belonging to source #2. Red crosses are classified as outliers. Observe that key-point classification segregates the key-points corresponding to the sources well.

4.2 Mixing System Estimation

Let $R_{2j}(\cdot) = T_{1j}(\cdot) - T_{2j}(\cdot) = R(\cdot, P_{2j})$ be a parametric representation (up to parameter vector P_{2j}) of a spatial distortion function. We use this notation since only relative spatial distortion between two distorted sources can be detected having no reference. Using the correspondences $\{p_k^{12} = (x_k^2, y_k^2), q_k^{12} = (u_k^2, v_k^2)\}_1^K$, we can now estimate the spatial distortion model. It consists of two distortion models; one for every source. We can find the models either by finding a joint model which is the union of the individual models, or by extracting the models sequentially. We choose the latter since it scales better when the number of sources increases. We use RANSAC [3] to achieve robust model estimation. The estimation steps are listed in Algorithm 1. Figure 3 visualizes the input to the spatial distortion estimation phase and its results.

The first RANSAC iteration recovers the first model and the second reveals the other. In addition to estimating the models we now have a classification of the correspondences according to their origins (up to source permutation) $\{(x_k^{21}, y_k^{21}), (u_k^{21}, v_k^{21})\}$ and $\{(x_k^{22}, y_k^{22}), (u_k^{22}, v_k^{22})\}$. Matches left after the estimation process are considered to be outliers. See Fig. 2 and Fig. 1.

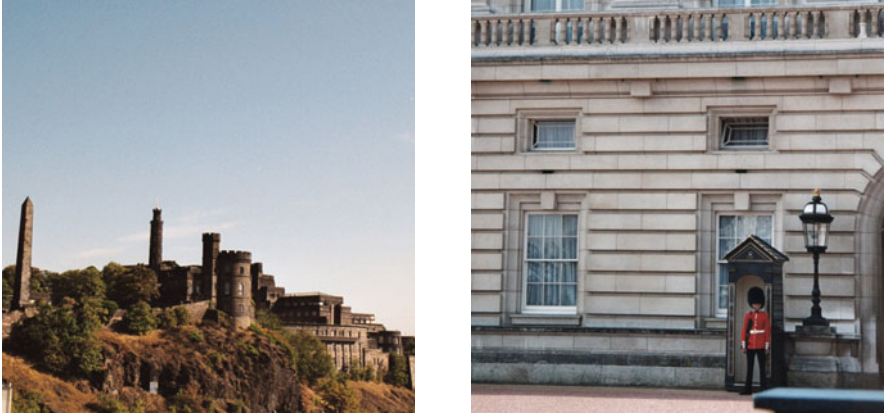


Fig. 2. Two source images used to demonstrate the proposed methods

Algorithm 1. Spatial Distortion Estimation

1. For $i = 1$ to Number of Sources
 - (a) Detect dominant model, based on correspondences, using RANSAC
 - (b) Tag RANSAC inliers as belonging to source i
 - (c) Remove RANSAC inliers from corresponding points' list.
-

The mixing coefficients are estimated in the same manner. This time, for each correspondence set $i \in \{1, 2\}$ we consider the ratio

$$A_{2j} = \frac{z_1(x_k^{2j}, y_k^{2j})}{z_2(u_k^{2j}, v_k^{2j})} \approx \frac{a_{1j}(\cdot)}{a_{2j}(\cdot)}$$

Since the attenuation model, a , is parametric, we can use RANSAC once again to find the parametric representation of the ratio A_{2i} .

We improve attenuation model estimation accuracy by taking advantage of the now known spatial distortion model to enrich the number of correspondences. Having estimated the spatial distortion model we can now transform the mixtures in a way that accurately aligns one of the sources. Having one of the sources aligned and using the sparse representation approach we add more samples to the attenuation model estimation process. Although this phase adds additional noise to the separation process, it is not significant since all other sources are not aligned and therefore add only random noisy samples. These samples are easily detected by the robust model fitting scheme.

This stage completes the mixing system estimation process, since we recover the relative spatial distortion, R_{2i} , and the relative attenuation function a_{2i} . To simplify notations we define $M_{ij} = (R_{ij}, a_{ij})$.

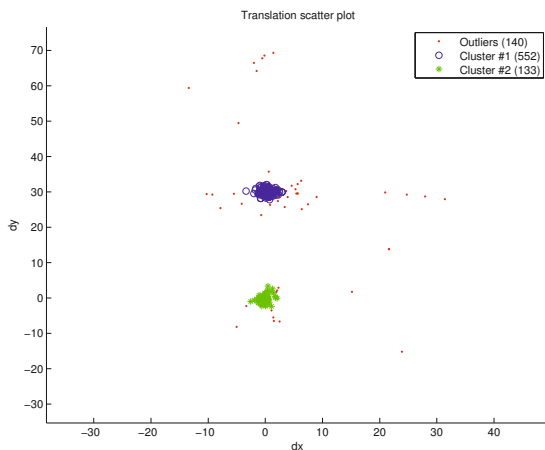


Fig. 3. Scatter plot of distortion vectors (dx, dy) of correspondences between two mixtures. In this example the spatial distortions are translations of the sources. The clusters correspond to two different distortions. One for each source. Clusters were extracted using Algorithm [1](#).



Fig. 4. Results of separation of the mixtures shown in Fig. [1](#) where mixtures were produced by spatial rotation and constant attenuation. (a) and (b) are the results of the separation process outlined in the text. (c) and (d) are results of using the real spatial distortion and attenuation models and separating the images as described in [4.3](#). This comparison shows that for this example separation quality is not limited by the mixing system estimation framework. Note that source order is not preserved.

4.3 Signal Separation

Having estimated the mixing unknowns, we can revert the mixing process. We use a simple matrix representation although variational methods [9] and quadratic programming methods [4] are suitable as well. Column stacking representation allows us to write the mixing system as:

$$\begin{bmatrix} \bar{z}_1 \\ \bar{z}_2 \end{bmatrix} = \begin{bmatrix} I & I \\ G_{21}(M_{21}) & G_{22}(M_{22}) \end{bmatrix} \begin{bmatrix} \bar{s}_1 \\ \bar{s}_2 \end{bmatrix}$$

The matrices G_{ij} can be easily constructed using the by-now-known to us models M_{ij} . We set the first block row of the matrix $G = \begin{bmatrix} I & I \\ G_{21} & G_{22} \end{bmatrix}$ to identity, since we have estimated only the relative distortion parameters. We arbitrarily choose \bar{z}_1 as the reference. Solving the linear system for $\begin{bmatrix} \bar{s}_1 \\ \bar{s}_2 \end{bmatrix}$ provides the desired separation. Although the dimension of the mixing matrix G is very large, it is also very sparse. The matrix is sparse as our mixing system is assumed to be anechoic. This observation limits the number of non-zero elements in every row of G to be not more than 2. Using various interpolation methods for sub-pixel accuracy may increase this number significantly. Such systems can be solved using methods like conjugate gradients or LSQR [14].

Solving the system may be a hard problem since it is usually under determined. A way to overcome this obstacle is by adding regularization terms, such as the Tikhonov regularization, into the matrix G . The regularization terms may add any prior knowledge constraints to the linear problem. Popular choices are smoothness and sparseness terms. When we have hard constraints (such as positivity) on the valid solutions, we may use the iterative projection onto convex sets (POCS) method to limit the solutions space. Using POCS we project the current solution onto the constrained solution space. We use this projection as an initial guess to the next solver iteration.

4.4 Generalizing to the $m \times n$ Scenario

We would like to use the same mixing system estimation and source separation, used for 2 sources and 2 mixtures, to separate m sources from n mixtures. For this purpose, we have to find the relative spatial distortion and attenuation M_{ij} between distorted sources w_{ij} and one of the mixtures, for simplicity \bar{z}_1 . Our only obstacle is the unknown source permutation. Let us consider the 3×3 scenario. The desired outcome of the estimation stage would be the mixing matrix G :

$$G = \begin{bmatrix} I & I & I \\ G_{21} & G_{22} & G_{23} \\ G_{31} & G_{32} & G_{33} \end{bmatrix} \quad (8)$$

However, the source estimation process does not provide us the second (column) index of the matrix block, and provides us with the mixing matrix G up to a

permutation of its row blocks. This is known as the permutation problem and was previously studied when separating sources in the frequency domain [15,19]. Since we are not interested in the real signal permutation we can leave the second block row permutation arbitrary (as in the 2×2 case) and permute the third block row accordingly.

We add a consistency constraint to find the correct permutation. We require mixtures z_i, z_j to align on their source components w_{ik}, w_{jk} originated in source s_k when inverse applying the inverse of operators G_{ik}, G_{jk} respectively. We may use this constraint to find pairs $\{(G_{ik}, G_{jk})\}$ that minimize some similarity criteria such as correlation or mutual information. We can use pairwise consistency as well as global consistency that measures the amount of agreement among all estimated models. If the models are too complex to simply apply the inverse transform, we may add another model estimation stage and find the relative distortion between z_i, z_j . By doing so we also require model consistency that requires the transformations $\bar{z}_i \rightarrow \bar{z}_1$ and $\bar{z}_j \rightarrow \bar{z}_i \rightarrow \bar{z}_q$ to be identical. The latter method can be used by itself to avoid image data similarity measurements.

The proposed method assumes that the estimated models significantly differ one from another, and that the mixed signals contain enough information from the sources to distinguish their contributions.

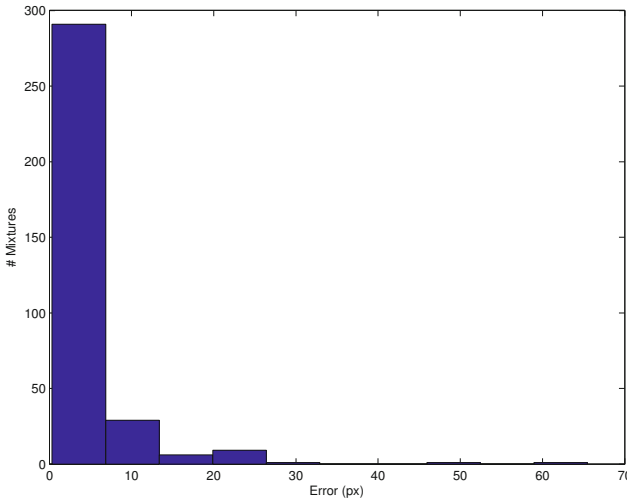


Fig. 5. Average spatial model error histogram. Mixtures were generated using random affine spatial distortions. The error per mixture was measured by comparing the estimated spatial distortion model with the model used. The models were compared pixel-wise using euclidean metric. The error has pixel units. We can observe that the vast majority of models were estimated up to a few pixels accuracy.

5 Results

We have tested the proposed framework on a variety of synthetic image mixtures. The separation results were evaluated both by visually comparing the separated

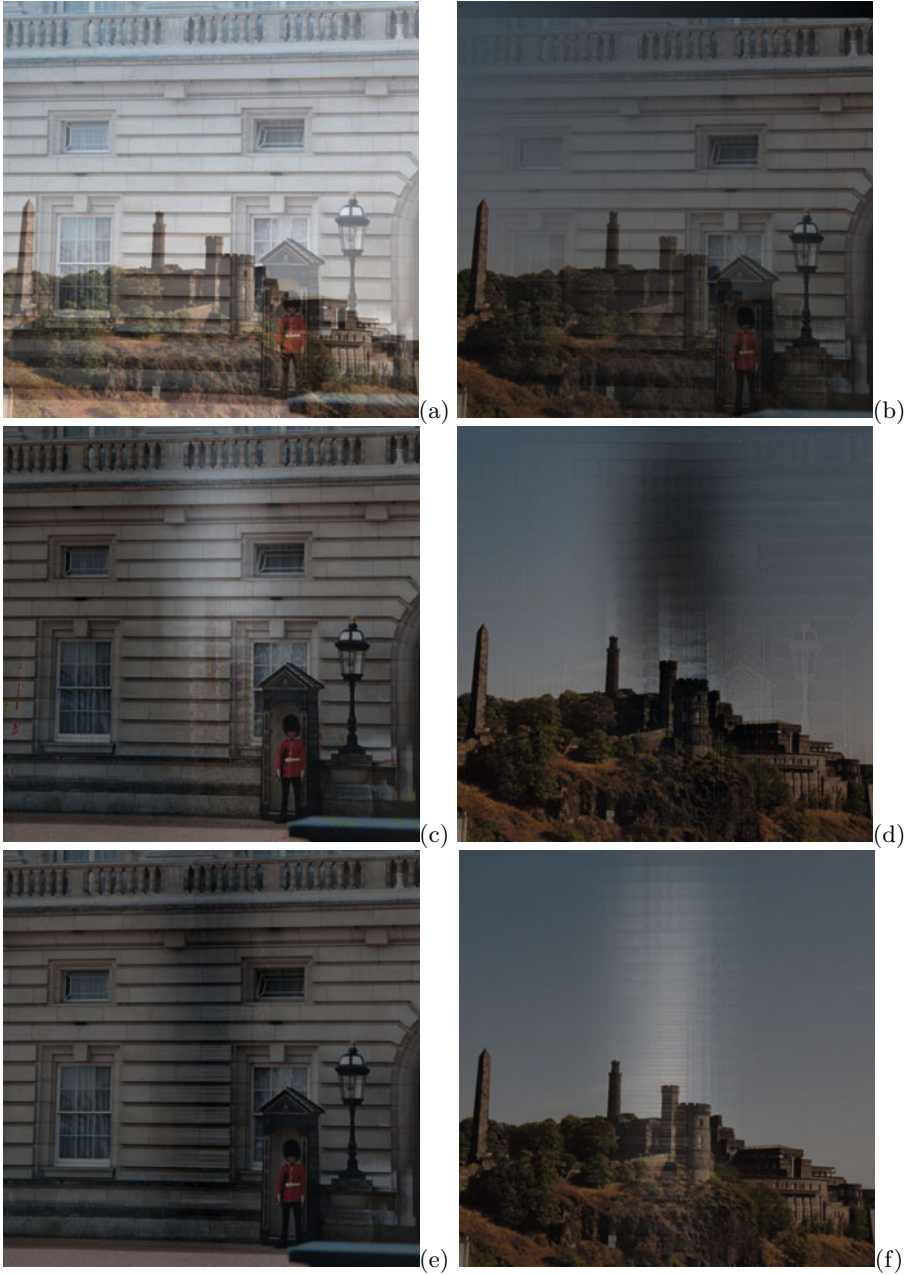


Fig. 6. Separation of spatially-varying non-instantaneous data. (a)-(b) Input mixtures with spatial translation and linear (fade) attenuation (c)-(d) Images unmixed using estimated models. (e)-(f) Images unmixed using known models.

images to their originals and by comparing the estimated mixing parameters to those that were used. Two such results are depicted in Fig. 4 and Fig. 6. Figure 5 shows the spatial distortion error distribution statistics, measured from 350 random mixtures. Mixture configurations that yielded low confidence, small consensus, model estimates were automatically detected and omitted from the chart. Such cases can be dealt with by using other set of parameters or even using different sparsification and correspondence detection techniques.

In general, reconstruction results are good when the model is estimated correctly. However small errors in the estimated model can lead to significant visual differences. As implied by the alignment algorithm, the quality of model estimation is determined by the quality and spread of the key-point correspondences and the sparseness of the problem. [Full sized results can be found at: <http://www.technion.ac.il/~albert/SinglePathBSS/samples.html>]

6 Conclusions

The proposed framework produces good results when the input signals meet a set of requirements such as sparseness, good key-point spread and parametric mixture model. However, since the algorithm is based on selected key-point correspondence, rather than dense (pixel to pixel) correspondence, significant information is unused. By combining the power of such sparse methods with dense correspondence, far more robust separation results can be achieved. The proposed method suffers from a reconstruction error induced by the model estimation error. For better separation a reconstruction scheme that allows small errors in model estimates should be applied.

Acknowledgement. Research supported by the Ollendorff Minerva Center for Vision and Image Sciences.

References

1. Bello, J.P., Daudet, L., Abdallah, S., Duxbury, C., Davies, M., Sandler, S.B.: A Tutorial on Onset Detection in Music Signals. *IEEE Transactions on Speech and Audio Processing* 13(5), 1035–1047 (2004)
2. Bronstein, M.M., Bronstein, A.M., Zibulevsky, M., Zeevi, Y.Y.: Blind Deconvolution of Images Using Optimal Sparse Representations. *IEEE Transactions on Image Processing* 14(6), 726–736 (2005)
3. Fischler, M.A., Bolles, R.C.: Random Sample Consensus: A Paradigm for Model Fitting With Applications to Image Analysis and Automated Cartography. *Commun. ACM* 24(6), 381–395 (1981)
4. Gai, K., Shi, Z., Zhang, C.: Blind Separation of Superimposed Images with Unknown Motions. In: *CVPR 2009*, pp. 1881–1888 (2009)
5. Jourjine, A., Rickard, S., Yilmaz, O.: Blind Separation of Disjoint Orthogonal Signals: Demixing n Sources from 2 Mixtures. In: *ICASSP 2000*, pp. 2985–2988 (2000)

6. Kaftory, R., Zeevi, Y.Y.: Probabilistic Geometric Approach to Blind Separation of Time-Varying Mixtures. In: Davies, M., James, C., Abdallah, S., Plumbley, M. (eds.) ICA 2007. LNCS, vol. 4666, pp. 373–380. Springer, Heidelberg (2007)
7. Kaftory, R., Zeevi, Y.Y.: Blind Separation of Images Obtained by Spatially-Varying Mixing System. In: ICIP 2008, pp. 2604–2607 (2008)
8. Kaftory, R., Zeevi, Y.Y.: Blind Separation of Position Varying Mixed Images. In: ICIP 2009, pp. 3913–3916 (2009)
9. Kaftory, R., Zeevi, Y.Y.: Blind Separation of Time/Position Varying Mixtures. CCIT Report #758, Technion - Dept. of Electrical Engineering (2010)
10. Kim, H.-G., Moreau, N., Sikora, T.: MPEG-7 Audio and Beyond: Audio Content Indexing and Retrieval. John Wiley & Sons, Chichester (2005)
11. Leordeanu, M., Hebert, M.: A Spectral Technique for Correspondence Problems Using Pairwise Constraints. In: ICCV 2005, pp. 1482–1489 (2005)
12. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* 60, 91–110 (2004)
13. Ogrady, P.D., Pearlmutter, B.A., Rickard, S.T.: Survey of Sparse and Nonsparse Methods in Source Separation. *International Journal of Imaging Systems and Technology* 15, 18–33 (2005)
14. Paige, C.C., Saunders, M.A.: LSQR: An Algorithm for Sparse Linear Equations And Sparse Least Squares. *ACM Trans. Math. Soft.* 8, 43–71 (1982)
15. Parra, L.C., Alvino, C.V.: Geometric Source Separation: Merging Convolutional Source Separation with Geometric Beamforming. *IEEE Transactions on Speech and Audio Processing* 10(6), 352–362 (2002)
16. Pedersen, M.S., Larsen, J., Kjems, U., Para, L.C.: A Survey of Convolutional Blind Source Separation Methods. *Springer Handbook of Speech Processing*. Springer, Berlin (2007)
17. Sakoe, H., Chiba, S.: Dynamic Programming Algorithm Optimization for Spoken Word Recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing* 26(1), 43–49 (1978)
18. Sarel, B., Irani, M.: Separating Transparent Layers through Layer Information Exchange. In: Pajdla, T., Matas, J.(G.) (eds.) ECCV 2004. LNCS, vol. 3024, pp. 328–341. Springer, Heidelberg (2004)
19. Smaragdakis, P.: Blind Separation of Convolved Mixtures in the Frequency Domain. *Neurocomputing* 22, 21–34 (1998)

Improving Gaussian Process Classification with Outlier Detection, with Applications in Image Classification

Yan Gao and Yiqun Li

Institute for Infocomm Research
Agency for Science, Technology and Research (A*STAR), Singapore

Abstract. In many computer vision applications for recognition or classification, outlier detection plays an important role as it affects the accuracy and reliability of the result. We propose a novel approach for outlier detection using Gaussian process classification. With this approach, the outlier detection can be integrated to the classification process, instead of being treated separately. Experimental results on handwritten digit image recognition and vision based robot localization show that our approach performs better than other state of the art approaches.

1 Introduction

Outliers refer to the data which do not fall into any learned classes in a classification system. Outlier detection is the identification of the unknown data or signal that a classification system is not aware of during training [1]. It is also usually referred to as novelty detection or abnormality detection. It is a common issue encountered in many computer vision applications, such as in robot vision [2], face recognition [3], and other image classification applications [4, 5]. Machine learning is a popular methodology for image classification. Using machine learning methods, outlier detection is usually treated as a one-class learning problem. Treating the given training samples as the ‘normal class’, a pre-assumed model is used to describe the normal class. In the test phase, a sample is classified as ‘normal’ or ‘abnormal’ by comparing it to the model. To model the normal class, various approaches have been explored, including clustering [6], nearest neighbor [7], mixture models [8], neural networks [4], self organizing maps (SOM) [9], and one class support vector machines (SVM) [5, 10, 11].

As the above works focus on the ‘one class’ problem, i.e., only concern about whether a new sample is normal or abnormal, it cannot solve the multi-class classification problem directly. There are plenty of applications that require both classification of a test sample into the existing classes as well as detection of outliers. When the application requires a multi-class classification, it needs 2 classification processes. One is the classification of normal and abnormal, namely, the outlier detection. The other is the classification of different normal classes. The method proposed in this paper is able to solve the multi-class classification problem with outlier detection simultaneously in one classification process. This

is because the proposed outlier detection method is inherently part of the Gaussian process classification process. Besides the outlier detection, the classifier can also refrain from making a decision when the confidence level is low as indicated by the winning class's probability estimate. In other words, the classifier will reject the unreliable classification results so that the classification can be more reliable to reduce the potentially high cost of misclassifications.

Although a few papers have discussed the multi-class problem in outlier detection, the objectives and problem scope are quite different. Masud et. al. proposed an outlier class detector within a decision tree or k nearest neighbor classifier [12]. It is specifically targeted for classification of data stream with possible concept-drift. The outliers must have some degree of coherence in order to form a novel class. In [3], a multi-class classifier with outlier detection is formed by combining multiple one-class classifiers. Multiple thresholds can be tuned to each of the one-class classifiers to improve the performances. However, solving a multi-class classification problem using one-class models will decrease the discrimination capability because the between-class variations is not considered. Hempstalk and Frank use a multi-class classification approach to solve outlier detection problem by assuming that training samples from the new classes are available [13]. Different from the above, our proposed approach solves a generic multi-class classification problem with more reliable detection of outliers in the test data. It does not require sample data from the abnormal class for training. During testing, the classifier will classify a test sample into one of the training classes, or detect it as an outlier, or refrain from making a decision if not sure. Gaussian process classification (GPC) produces a probabilistic classifier which includes both prediction of probabilities of a sample belonging to the training classes, as well as a covariance matrix of the predicted probabilities. We make use of the covariance matrix for outlier detection. The proposed approach is evaluated on 2 benchmark datasets for handwritten digit recognition and robot localization and shows promising results.

2 Multi-class Gaussian Process Classification

We first give a brief introduction to multi-class Gaussian process classification. For more details, the readers are referred to [14]. For a multi-class problem, we are given a set of input vectors $X = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)^T$ and a target vector $\mathbf{y} = (y_1^1, \dots, y_n^1, y_2^2, \dots, y_n^2, \dots, y_1^C, \dots, y_n^C)^T$, where n is the number of input vectors, C is the number of classes, $y_i^c = 1$ if the i^{th} input vector belongs to the c^{th} class, and it is all zero otherwise.

In order to make inference, a vector of latent function values \mathbf{f} is introduced. $\mathbf{f} = (f_1^1, \dots, f_n^1, f_2^2, \dots, f_n^2, \dots, f_1^C, \dots, f_n^C)^T$. It is assumed that the C latent processes are uncorrelated. A prior over the latent function is specified. It follows a normal distribution with a mean of 0 and a covariance matrix K :

$$\mathbf{f} \sim \mathcal{N}(0, K) \quad (1)$$

The covariance matrix K is block diagonal with sub-matrices K_1, \dots, K_C on the diagonal. The covariance matrix for each of the C classes is defined by its own covariance function:

$$K_{c(i,j)} = k_c(\mathbf{x}_i, \mathbf{x}_j), \quad i, j = 1, \dots, n \tag{2}$$

The target vector \mathbf{y} is related to the latent function vector \mathbf{f} by:

$$p(y_i^c | \mathbf{f}_i) = \pi_i^c = \frac{\exp(f_i^c)}{\sum_{c'} \exp(f_i^{c'})} \tag{3}$$

where $\mathbf{f}_i = (f_i^1, \dots, f_i^C)^T$.

The posterior $p(\mathbf{f} | X, \mathbf{y})$ is proportional to the joint probability $p(\mathbf{y} | \mathbf{f})p(\mathbf{f} | X)$, based on the Baye's theorem. The log of the un-normalized posterior is shown to be:

$$\Phi(\mathbf{f}) = -\frac{1}{2} \mathbf{f}^T K^{-1} \mathbf{f} + \mathbf{y}^T \mathbf{f} - \sum_{i=1}^n \log \left(\sum_{c=1}^C \exp f_i^c \right) - \frac{1}{2} \log |K| - \frac{Cn}{2} \log 2\pi \tag{4}$$

As the posterior is not analytically tractable, the Laplace approximation is used to give a Gaussian approximation $q(\mathbf{f} | X, \mathbf{y})$ to the posterior $p(\mathbf{f} | X, \mathbf{y})$. To do this, a second order Taylor expansion of $\log p(\mathbf{f} | X, \mathbf{y})$ is needed around the maximum of the posterior. Denote the value that maximizes the posterior as $\hat{\mathbf{f}}$,

$$\hat{\mathbf{f}} = \arg \max_{\mathbf{f}} \Phi(\mathbf{f}) \tag{5}$$

and it is found using the Newton's method.

To predict the label of a new input \mathbf{x}_* , the posterior distribution $q(\mathbf{f}_* | X, \mathbf{y}, \mathbf{x}_*)$ is given by

$$q(\mathbf{f}_* | X, \mathbf{y}, \mathbf{x}_*) = \int p(\mathbf{f}_* | X, \mathbf{x}_*, \mathbf{f}) q(\mathbf{f} | X, \mathbf{y}) d\mathbf{f} \tag{6}$$

Both $p(\mathbf{f}_* | X, \mathbf{x}_*, \mathbf{f})$ and $q(\mathbf{f} | X, \mathbf{y}) d\mathbf{f}$ are Gaussian. Therefore $q(\mathbf{f}_* | X, \mathbf{y}, \mathbf{x}_*)$ is also Gaussian. Its mean is given by

$$\mathbb{E}_q[\mathbf{f}(\mathbf{x}_* | X, \mathbf{y}, \mathbf{x}_*)] = Q_*^T K^{-1} \hat{\mathbf{f}} = Q_*^T (\mathbf{y} - \hat{\boldsymbol{\pi}}) \tag{7}$$

where

$$Q_* = \begin{pmatrix} \mathbf{k}_1(\mathbf{x}_*) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{k}_2(\mathbf{x}_*) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{k}_C(\mathbf{x}_*) \end{pmatrix} \tag{8}$$

where $\mathbf{k}_c(\mathbf{x}_*)$ is the vector of covariances between the test point and each of the training points, evaluated by class c 's covariance function.

The covariance is given by

$$\begin{aligned} \text{cov}_q(\mathbf{f}_* | X, \mathbf{y}, \mathbf{x}_*) &= \Sigma + Q_*^T K^{-1} (K^{-1} + W)^{-1} K^{-1} Q_* \\ &= \text{diag}(\mathbf{k}(\mathbf{x}_*, \mathbf{x}_*)) - Q_*^T (K + W^{-1})^{-1} Q_* \end{aligned} \tag{9}$$

where Σ is a diagonal $C \times C$ matrix with $\Sigma_{cc} = k_c(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_c^T(\mathbf{x}_*)K_c^{-1}\mathbf{k}_c(\mathbf{x}_*)$, and $\mathbf{k}(\mathbf{x}_*, \mathbf{x}_*)$ is a vector of covariances, whose c^{th} element is $k_c(\mathbf{x}_*, \mathbf{x}_*)$.

The marginal likelihood $\log p(\mathbf{y}|X, \theta)$ can be similarly approximated as

$$\begin{aligned} \log p(\mathbf{y}|X, \theta) &\simeq \log q(\mathbf{y}|X, \theta) \\ &= -\frac{1}{2}\hat{\mathbf{f}}^T K^{-1}\hat{\mathbf{f}} - \sum_{i=1}^n \log\left(\sum_{c=1}^C \exp \hat{f}_i^c\right) - \frac{1}{2} \log |I_{Cn} + W^{\frac{1}{2}}KW^{\frac{1}{2}}| \end{aligned} \quad (10)$$

The marginal likelihood can be used to tune the parameters of the covariance functions which are also known as the hyperparameters of the model.

3 Outlier Detection in Gaussian Process Classification

To detect outliers under the Gaussian process (GP) classification framework, the covariance in prediction plays an important role. Recall from the previous section that the prediction made by GP classification is characterized by a mean (Eq. (7)) and a covariance matrix (Eq. (9)). In Gaussian process, the variance in prediction is large when the new sample is out of the support of the training samples. (See e.g., illustrations in [15, 16]) The total variance in the covariance matrix is an indicator of how familiar the classifier is about a particular test sample. We propose to use the determinant of the covariance matrix as the measure of novelty, and the rule is given by:

$$\begin{cases} l(\mathbf{x}_*) = -1 & \text{if } \det(\text{cov}_q(\mathbf{f}_*|X, \mathbf{y}, \mathbf{x}_*)) > t \\ l(\mathbf{x}_*) = \arg \max_c p(y_*^c|\mathbf{f}_*) & \text{otherwise} \end{cases} \quad (11)$$

where $l(\mathbf{x}_*)$ refers to the label of the sample \mathbf{x}_* , and -1 is used as the label of the outliers. As in Eq. (3), $p(y_*^c|\mathbf{f}_*) = \frac{\exp(f_*^c)}{\sum_{c'} \exp(f_*^{c'})}$. Alternatively, we can sort the test data according to the novelty measure (which is the determinant of covariance matrix) in a descending order, and classify a certain amount of test samples with largest novelty measures as outliers, if such information is given by prior knowledge.

The choice of using the determinant of the covariance function as the novelty measure is not just heuristic. Recall that the determinant is equal to the sum of the eigenvalues of the covariance matrix. We know that the eigenvalues of the covariance matrix indicate the portions of variance that are explained by the principal components (see principal component analysis [17]). Therefore the sum of all eigenvalues reflects the total variance involved with the particular prediction. To illustrate the idea, a toy data is designed. It consists of three 2-d Gaussian clusters to form the three classes. A few mis-labeled samples are also simulated. We use the squared exponential covariance function,

$$k(\mathbf{x}_i, \mathbf{x}_j) = \alpha \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\beta^2}\right) \quad (12)$$

The hyperparameters α and β are determined by optimizing the likelihood function in Eq. (10). In Fig. 1, we show a contour plot of the novelty measure

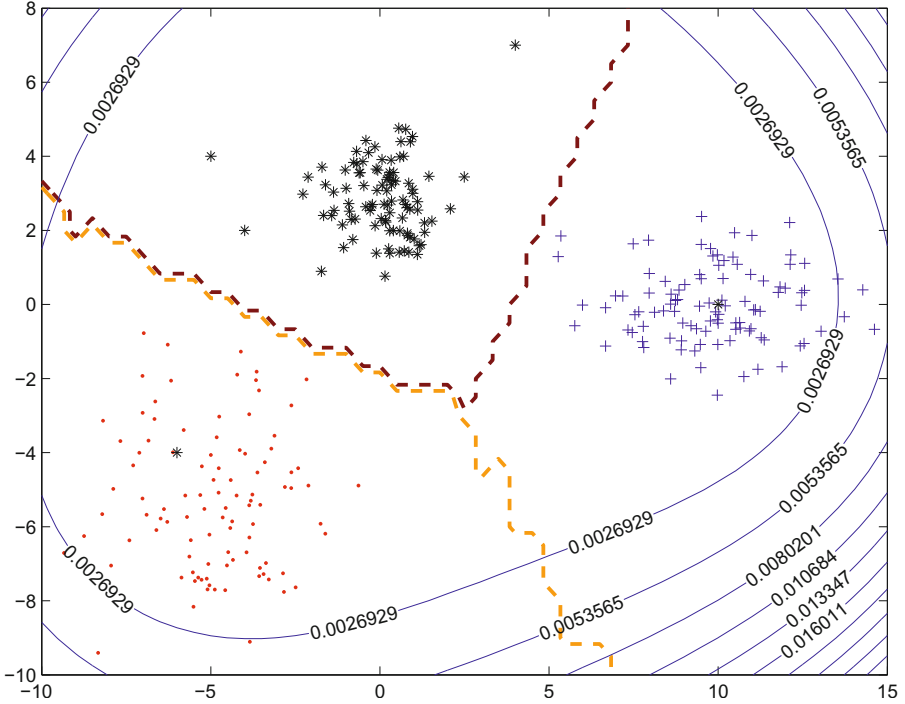


Fig. 1. Contour plot of the proposed novelty measure and partition of the input space into different classes

$\det(\text{cov}_q(\mathbf{f}_*|X, \mathbf{y}, \mathbf{x}_*))$ in the input space. It is observed that the proposed novelty measure gets larger when moving away from the cluster centers. In the meantime, the GP classifier also partitions the input space into the three training classes.

4 A Well-Rounded Classifier

With the proposed method for outlier detection, Gaussian process classification may offer advantages over other alternative classifiers to many real problems. Recall that GP classification produces a probabilistic prediction. The probability of a test sample belonging to all training classes are explicitly obtained by the classifier (Eq. (3)). As mentioned in [14], the probability of the test sample belonging to the winning class can be used to reject unreliable predictions. If it is low, it shows that the classifier is not confident in classifying the test sample into a particular class. In this case, it might be advantageous to refrain from making a decision than making a wrong decision with high probability. This is known as the reject option in classification. To show how it works, we also plot the winning classes' probabilities for the three class classification problem in the previous section in Fig. 2. It is observed that the winning classes' probabilities

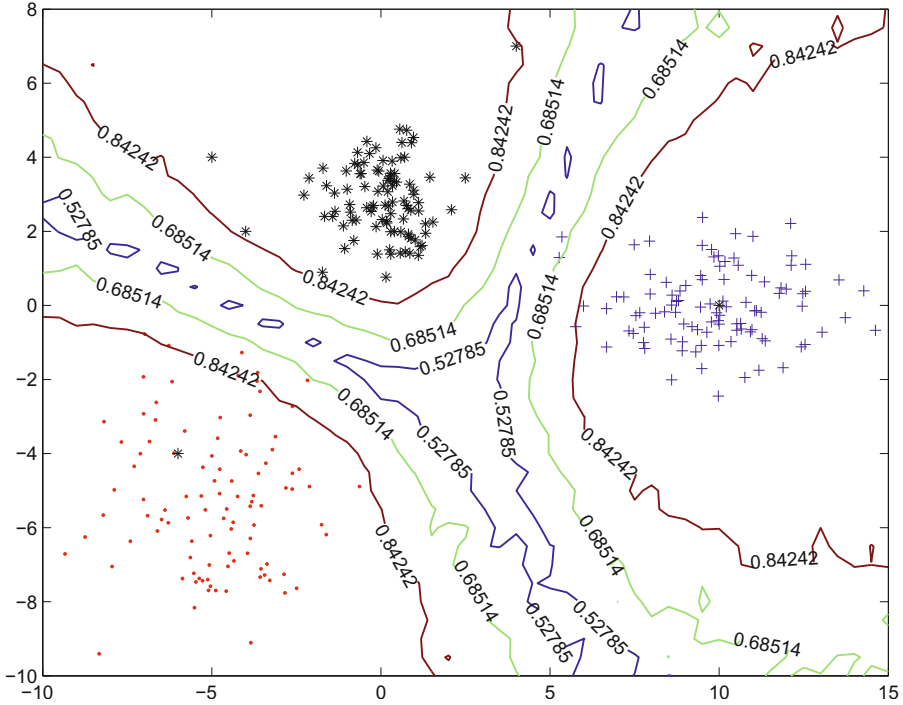


Fig. 2. Contour plot of the winning classes' probabilities

are smaller in between two training classes where it is most likely to make wrong predictions from a Bayesian point of view.

With the capabilities of detecting outliers and rejecting unreliable predictions, Gaussian process classification is well suited for some applications such as the robot localization application discussed in section 5.2.

5 Experiments

We implement the multi-class GP classification using Laplace approximation as outlined in section 2 and the outlier detection method in section 3 in Matlab. We compare the proposed outlier detection scheme with one class support vector machines which have been shown to be a state of the art for outlier detection, and have been popularly used in various applications [5, 11, 18, 19, 20]. The basic idea of one-class SVM is to find an enclosing boundary for the normal samples in the kernel space. The classification performance is compared to multi-class support vector machines (SVM). In SVM, the Gaussian radial basis function (RBF) is used as the kernel. The Gaussian width is set to the mean of pairwise distances among training samples. For one class SVM, the parameter ν that is used to control the percentage of training data that is allowed outside the enclosing boundary is set to 5%. For multi-class SVM, the cost parameter for controlling tradeoff between complexity and training accuracy is set to 100 [21].



Fig. 3. Sample images from the USPS handwritten digit image dataset (left) and the alphabet and digit (AlphaDigs) image dataset (right)

5.1 Handwritten Digits Recognition

We first experiment on handwritten digit recognition and consider alphabet images as outliers. The USPS handwritten digit dataset is used. It consists of 4649 training images and 4649 test images of 10 digit classes¹. The raw image intensity is used as the image feature. For Gaussian process classification, the square exponential covariance function (Eq. (12)) is used. The hyperparameters are tuned by maximizing the marginal likelihood and we simply adopt the values according to that in [14]. On the test data, an overall accuracy of 96.5% is achieved, which is consistent with that reported in [14]. To evaluate the proposed outlier detection scheme, we test the classifier trained on the USPS data on a completely different alphabet and digit (AlphaDigs) dataset². It consists of 39 images for each of the 10 digit classes and 26 alphabet classes. Sample images from both the USPS dataset and the AlphaDigs dataset are shown in Fig. 3.

For outlier detection, the images in the AlphaDigs dataset are sorted according to the novelty measure that is used to determine if a sample is an outlier. In the proposed method, it is the determinant of the covariance matrix in a descending order. For one-class SVM, it is the distance to the enclosing hyperplane (with distance outside the hyperplane being positive) in a descending order. A certain amount of test samples with largest novelty measures are then classified as the outliers. We evaluate the outlier detection performance using the receiver operating characteristic (ROC) curve. The ROC curve is a plot of sensitivity (true positive rate) against specificity (false positive rate). The true positive rate is equal to the number of alphabet images that are correctly detected as outliers divided by the total number of alphabet images. The false positive rate is the number of digit images that are wrongly detected as outliers divided by the total number of digit images. The threshold is set at various values in order to obtain a set of points to plot the curves. From Fig. 4, the proposed outlier detection scheme clearly out-performs that of one-class SVM.

It is also worth mentioning the relative classification performance of Gaussian process classification and multi-class SVM. The results are shown in Table 1. It is observed that while Gaussian process classification and SVM give comparable results on the test data from USPS dataset, the former gives a much better result

¹ Available at <http://www.gaussianprocess.org/gpml/data/>

² Available at <http://cs.nyu.edu/~roweis/data.html>

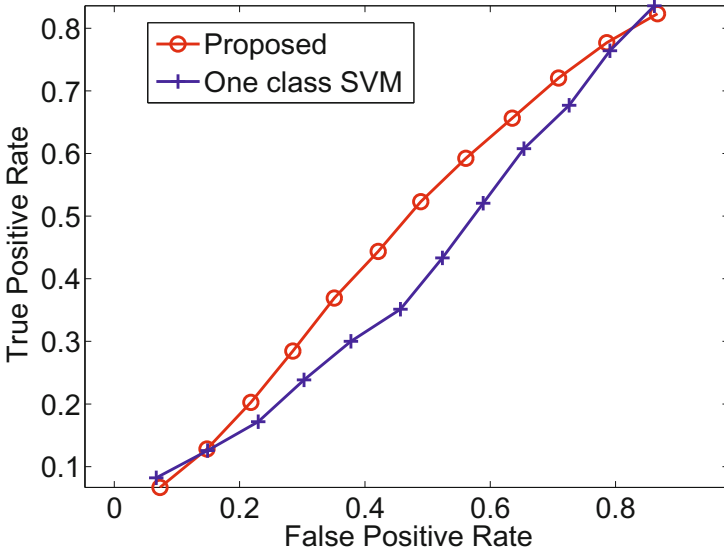


Fig. 4. ROC curves for outlier detection using the proposed method and one class SVM

on the digit images from the AlphaDigs dataset. Since the AlphaDigs dataset is independently collected, and the classifiers are trained on the USPS data, it is considered a more difficult dataset compared with the test set from the USPS data. This shows that Gaussian process classification has a better generalization capability on a different dataset as compared with that of SVM. Note that on the AlphaDigs dataset, only the digit images are used to evaluate the classification performance. This is because the AlphaDigs dataset is dominated by outliers (alphabet images), thus including them will make the evaluation of classification accuracy heavily biased towards detecting more outliers.

Table 1. Classification accuracy comparison using multi-class Gaussian process classification (mcGPC) and multi-class support vector machine (mcSVM)

	mcGPC	mcSVM
On USPS test dataset	96.5%	97.2%
On digit images from the AlphaDigs dataset	72.31%	62.31%

5.2 Localization of Mobile Robots

In this experiment we show the capabilities of Gaussian process classification in terms of both outlier detection and rejection of unreliable predictions. In a robot localization problem, a set of training sequences of various locations are acquired for training a classifier. The classifier answers the question “where am I” when presented with a test sequence. The test sequence may contain locations that

were not imaged in the training sequences. These locations should be classified as the ‘unknown’ class. In addition, the classifier may refrain from making a decision when it is not confident about a particular prediction. Therefore, there are two types of uncertainties faced by the classifier when trying to classify a test sample into the training classes. One is that the new sample is not similar to any of the training classes and therefore it is likely to come from a new class. The second type is that the new sample is equally similar to two or more training classes and therefore cannot be classified with strong confidence.

The training and validation data are from the IDOL2 database [22]. The image sequences in the database are acquired using the MobileRobots PowerBot robot platform. The training sequence consists of 1034 image frames of 5 classes according to the robot’s topological location, namely, one-person office(BO), corridor(CR), two-persons office(EO), kitchen(KT), and printer area(PA). The test sequence consists of 1690 image frames classified into 6 classes, 5 of which are the same as those of the training sequence, and one additional unknown(UK) class corresponding to the additional rooms that are not imaged previously. The test sequence is acquired 20 months after the training sequence. For more details please refer to [23].

Gradient based features are chosen as the robot is in indoor environment with strong edge characteristics. Each training image in the training sequence is described by normalized Gaussian derivatives on the L component of the LAB color space. 5 partial derivatives ($L_x, L_y, L_{xx}, L_{yy}, L_{xy}$) are computed and quantized into 32 bins built by k -means. A three-tier spatial pyramid of histograms is then obtained on each image. Each image is represented by a 672 dimensional feature vector.

Using Gaussian process classification, an overall classification accuracy of 55.8% is obtained (the unknown class is treated equally as the training classes in calculation of classification accuracy). Note that the test data include about 20% outliers which are from locations that were not imaged in the training sequence. Without outlier detection, these 20% outliers will be classified into one of the training classes and this explains the low overall classification accuracy. Fig. 5 (a) shows the improvement in classification accuracy if a certain amount of samples are classified as outliers based on the proposed novelty measure. If the prior knowledge that about 20% outliers are present, the classification accuracy is improved to about 59%. We compare the performance with that of multi-class support vector classification with outlier detection by one class SVM. As addressed earlier, in this case, the classification process is independent from the outlier detection. Without outlier detection, the classification accuracy is 56.45%, which is slightly better than that of Gaussian process classification. With outlier detection using one class SVM, the classification accuracy also improves with a certain amount of samples detected as outliers. But the improvements are not as much as that of the proposed method, and we observe a narrower window before the classification accuracy drops below the baseline (Fig. 5(a)).

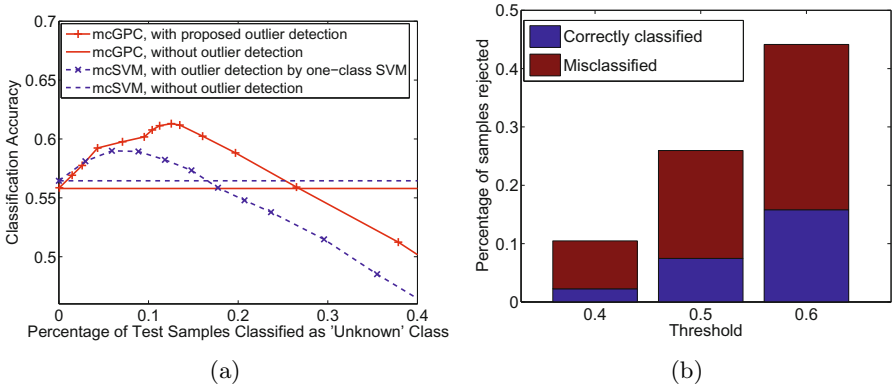


Fig. 5. (a) Classification accuracy using multi-class GPC and multi-class SVM, with or without outlier detection. (b) Rejection of unreliable predictions.

Further, if we make use of the rule proposed in section 4 to reject unreliable predictions at a threshold of winning classes’s probability exceeding 0.4, 0.5, and 0.6, the classification accuracy further improves to 62.19%, 66.51%, and 71.61%, respectively. Figure 5(b) shows the relative proportions of correctly classified and misclassified in the rejected samples. It is observed that it is dominated by misclassified samples, showing that the reject rule is useful in reject unreliable predictions. If rejection of a sample has a lower cost than misclassifying a sample, the reject rule could help reduce the overall cost of the classification. For example, if the cost of correctly classifying a sample is 0, wrongly classifying a sample is 1, and making no decision about a sample is 0.5, the savings in cost by rejecting at the three thresholds are 50.5, 93.5, and 106, respectively.

6 Conclusion

In this paper, we explore the outlier detection capability of a Gaussian process classifier. It is shown that the determinant of the covariance matrix from the output of the Gaussian process classifier is a good measure of how novel a test sample is compared to the training samples. With this discovery, Gaussian process classifier, as a probabilistic classifier, is able to handle both outlier detection and rejection of unreliable predictions. Experiments on two practical applications show the advantages of the Gaussian process classification with outlier detection.

References

1. Markou, M., Singh, S.: Novelty detection: a review-part 1: statistical approaches. *Signal Processing* 83, 2481–2497 (2003)

2. Automatic Outlier Detection: A Bayesian Approach. In: 2007 IEEE International Conference on Robotics and Automation (2007)
3. Tax, D.M.J., Duin, R.P.W.: Growing a multi-class classifier with a reject option. *Pattern Recogn. Lett.* 29, 1565–1570 (2008)
4. Singh, S., Markou, M.: An approach to novelty detection applied to the classification of image regions. *IEEE Trans. on Knowl. and Data Eng.* 16, 396–407 (2004)
5. Lukashevich, H., Nowak, S., Dunker, P.: Using one-class svm outliers detection for verification of collaboratively tagged image training sets. In: ICME 2009, pp. 682–685 (2009)
6. Loureiro, A., Torgo, L., Soares, C.: Outlier detection using clustering methods: a data cleaning application. In: *Proceedings of the Data Mining for Business Workshop* (2004)
7. Ramaswamy, S., Rastogi, R., Shim, K.: Efficient algorithms for mining outliers from large data sets. *SIGMOD Rec.* 29, 427–438 (2000)
8. Lauer, M.: A mixture approach to novelty detection using training data with outliers. In: Flach, P.A., De Raedt, L. (eds.) *ECML 2001. LNCS (LNAI)*, vol. 2167, pp. 300–311. Springer, Heidelberg (2001)
9. Xing, H., Wang, X., Zhu, R., Wang, D.: Application of kernel learning vector quantization to novelty detection, pp. 439–443 (2008)
10. Muñoz, A., Moguerza, J.M.: One-class support vector machines and density estimation: The precise relation. In: Sanfeliu, A., Martínez Trinidad, J.F., Carrasco Ochoa, J.A. (eds.) *CIARP 2004. LNCS*, vol. 3287, pp. 216–223. Springer, Heidelberg (2004)
11. Chen, Y., Zhou, X.S., Huang, T.: One-class Svm for Learning in Image Retrieval, vol. 1, pp. 34–37 (2001)
12. Masud, M.M., Gao, J., Khan, L., Han, J., Thuraisingham, B.: Integrating novel class detection with classification for concept-drifting data streams. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds.) *ECML PKDD 2009. LNCS*, vol. 5782, pp. 79–94. Springer, Heidelberg (2009)
13. Hempstalk, K., Frank, E.: Discriminating against new classes: One-class versus multi-class classification. In: Wobcke, W., Zhang, M. (eds.) *AI 2008. LNCS (LNAI)*, vol. 5360, pp. 325–336. Springer, Heidelberg (2008)
14. Rasmussen, C.E., Williams, C.K.I.: *Gaussian Processes for Machine Learning*. The MIT Press, Cambridge (2006)
15. Grochow, K., Martin, S.L., Hertzmann, A., Popović, Z.: Style-based inverse kinematics. In: *ACM Special Interest Group on Graphics and Interactive Techniques Conference (SIGGRAPH)*, pp. 522–531 (2004)
16. Lawrence, N.D.: Gaussian process latent variable models for visualisation of high dimensional data. In: *Advances in Neural Information Processing Systems*, vol. 16 (2004)
17. Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern Classification*, 2nd edn. Wiley, Chichester (2001)
18. Clifton, L.A., Yin, H., Clifton, D.A., Zhang, Y.: Combined support vector novelty detection for multi-channel combustion data. In: *ICNSC*, pp. 495–500 (2007)
19. Tax, D.M.J., Ypma, A., Duin, R.P.W.: *Support Vector Data Description Applied to Machine Vibration Analysis* (1999)
20. Heller, K.A., Svore, K.M., Keromytis, A.D., Stolfo, S.J.: One class support vector machines for detecting anomalous windows registry accesses. In: *Proc. of the Workshop on Data Mining for Computer Security* (2003)

21. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines Software (2001), <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
22. Luo, J., Pronobis, A., Caputo, B., Jensfelt, P.: Incremental learning for place recognition in dynamic environments. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007), San Diego, CA, USA (2007)
23. Caputo, B., Pronobis, A., Jensfelt, P.: Overview of the clef 2009 robot vision track. In: CLEF working notes 2009, Corfu, Greece (2009)

Robust Tracking Based on Pixel-Wise Spatial Pyramid and Biased Fusion

Huchuan Lu¹, Shipeng Lu¹, and Yen-Wei Chen^{1,2}

¹ School of Information and Communication Engineering,
Dalian University of Technology, Dalian, China

² College of Information Science and Engineering,
Ritsumeikan University, Kusatsu, Japan

Abstract. We propose a novel tracking algorithm for the balance between stability and adaptivity as well as a new online appearance model. Since the update error is inevitable, we present three tracking modules, i.e., reference model, soft reference model and adaptive model, and fuse them using biased multiplicative formula. These three contributors are built through the same appearance model with different update rate. The appearance model, Pixel-wise Spatial Pyramid, employs pixel feature vectors instead of SIFT vectors, to combine several pixel characteristics. In particular, the reserved pixel feature vectors are used to create a new codebook together with the earlier codebook. A hybrid feature map consisting of the reserved pixel vectors and anti-part of previous hybrid feature map is built to represent the new target map. Experimental results show that our approach tracks the object with drastic appearance change, accurately and robustly.

1 Introduction

Visual object tracking is one of the well-known problems in the computer vision community. Tracking intrinsically focuses on comparison problem: In general, tracking system can be thought of similarity metric-based algorithm or classification-based algorithm. Many similarity metric-based trackers have been proposed, such as probabilistic models using mean-shift [1,2] or particle filtering [3], IVT [4] and FragTrack [5]. Classification-based algorithms [6,7,8], meaning to optimally discriminate the object from the current background, perform well on various challenging conditions. Our tracker based on pixel-wise spatial pyramid and biased multiplicative formula falls into the first category.

To deal with the significant appearance variations in the video sequences, due to the pose variation, shape deformation, scale change, illumination change, camera motion, and occlusions, tracking algorithm should be adaptive through the online update. The most of previous online tracking algorithms using a self-learning policy, i.e., the tracker relies on its own predictions, unfortunately faces a severe drifting problem. This trouble can be explained by the stability-plasticity dilemma [9]: If the tracker is built only with the initial information, it is the least error-prone to drift but can not survive undergoing appearance and viewpoint

changes. On the contrary, the self-learning online tracker is highly adaptive but easily drifts. Some methods have been proposed to find the trade-off between adaptivity and stability.

Grabner *et al.* [10] developed tracking as a semi-supervised learning problem using online boosting. It has shown to be less susceptible to drifting while adaptive, but it keeps the non-optimal prior. Recently, they [11] have advanced the tracker by extending the semi-supervised learning approach with adaptive priors, making it robust to track multiple similar objects.

Babenko *et al.* [8] successfully used online multiple instance learning to overcome the ambiguities of bounding boxes during tracking, and got the state-of-the-art results. Santner *et al.* [12] tackled this problem by combining several complimentary trackers operating at different timescales.

We also address the robustness and adaptivity of online appearance-based tracking regarding the reliability or credit of the tracking model in this paper. The underlying assumption is that the update can not be absolutely correct and drifting risk always exists, because the supposed objects used to update are more or less wrong, with ambiguity or label jitter. For the initial appearance model, it is the most reliable one during the tracking period, then different update rate of the model means different confidence, i.e., the more modified the model is, the less trustworthy it is. Specially, we make use of the initial (stable) appearance model, a soft stable appearance model and a novel adaptive appearance model, eventually fuse them using biased multiplicative principle (Fig. 1). Note that the appearance models are built using the same method, only with different adaptivity rates.

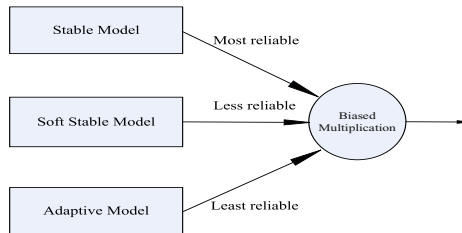


Fig. 1. Fusion of the three models

There is no doubt that effectively modeling appearance variations plays a critical role in visual tracking. Many researchers [1, 2, 3, 4, 13, 14] focus on the design of appearance model to strengthen the discriminability. Porikli *et al.* [13] proposed a covariance matrix descriptor for characterizing the appearance of an object to capture both statistical and spatial properties of object appearance. In particular, the covariance matrix descriptor offers a principal way to fuse several features through pixel feature vector style. Meanwhile, Arif *et al.* [14] employed the individual pixel feature vectors as observation in the KPCA eigenspace to create a pixel-wise appearance model which is robust to noise and occlusions, whereas previous approaches used vectorized image regions as observation.

Bag of words (bag of features) [15,16] representations have become popular for content based image classification and object localization owing to their simplicity and good performance. The main idea is to treat images as loose collections of independent local features, using the cluster label distribution in feature space as a characterization of the image. However, because these methods disregard all information about the spatial layout of the features, they have markedly limited descriptive ability. Lazebnik *et al.* [17] developed “spatial pyramid”, a simple and computationally efficient extension of the orderless bag of features image representation, and gained significantly improved performance on challenging scene categorization tasks.

The second contribution of this paper is to present a novel online learning tracker with a new appearance model and an update scheme designed for the model. We build “spatial pyramid” using pixel-wise feature vectors in the region of interest. Pixel-wise feature vector consists of several individual pixel-features. During the process of update, a codebook built by K-means is carefully modified through the distance-based scheme. We generate a hybrid feature map, i.e., the new valuable information in the current frame and the cumulative information from previous images, to absorb the essence and reject the dross as much as possible.

We briefly depict an overview of our method in section 2, describe the online learning approach with pixel-wise spatial pyramid for visual tracking in section 3, give a detailed analysis of model fusion rule in section 4, and discuss the experimental evaluation in section 5, followed by conclusions and future work in the last section.

2 Overview of the Method

This section gives an overview of our tracking system, which is summarized in Fig. 2. Our goal is to make the tracking algorithm to be adaptive to drastic appearance changes and recoverable from drifting. Therefore, we elaborately develop a discriminative appearance model and considerate update approach, then make the tracker more robust and stable using the biased multiplicative principle. The functional result is to find the biased balance point among the three tracking components with dissimilar update.

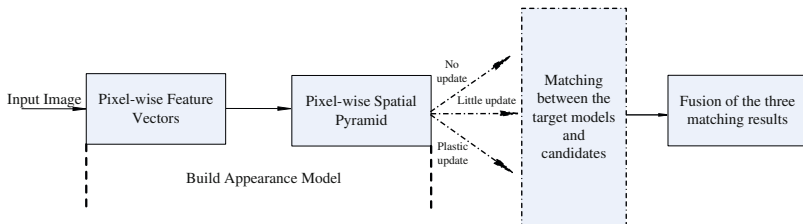


Fig. 2. An overview of our tracking algorithm

Pixel-wise feature vector, combining manifold traits instead of using one kind of feature, could be made more discriminative. We use these pixel-wise features instead of local features (e.g., SIFT [18] or HOG [19]) in the building of “spatial pyramid” to employ multiple features. The tracker’s performance proves that it is feasible and effective.

We introduce the model update rate α , where α denotes the level of the update of the appearance representation, i.e., the percentage of the reserved current pixel feature vectors unlike the adaptivity rate in [12]. The adaptivity rate in [12] denotes the number of frames a tracker needs to fully adapt to appearance changes. (i) The most reliable reference information without update does not suit appearance changes with $\alpha = 0$. (ii) The soft stable appearance model with mid-update, to some extent, copes with the appearance variations with $0.1 < \alpha < 0.4$. (iii) Frame-to-frame online tracker with a moderate ratio of update fits the appearance changes as well as possible from the premise of eliminating the noise information, with $0.4 < \alpha < 0.9$. In this paper, the soft stable model with $\alpha = 0.15$ is updated every three frames. The online model is updated each frame, with α determined by an empirical threshold.

3 Our Online Learning Tracker

3.1 Sequential Inference Model

Our on-line tracker meets the Bayesian Inference for visual tracking, which is a Markov model with hidden state variables. Using Bayes’ theorem, the tracking equation can be written as follows:

$$p(X_t/D_t) \propto p(I_t/X_t) \int p(X_t/X_{t-1})p(X_{t-1}/D_{t-1})dX_{t-1} \quad (1)$$

To benefit the building of appearance model, an affine motion sampling model as in [4] is used to attain the candidates. Hidden state variables X_t denote the affine motion parameters by six parameters $X_t = \{x_t, y_t, \theta_t, s_t, \alpha_t, \phi_t\}$, and $x_t, y_t, \theta_t, s_t, \alpha_t, \phi_t$ denote x, y translation, rotation angle, scale, aspect ratio, and skew direction at time t , I_t describes the observed image at t frame, and $D_t = \{I_1, I_2, \dots, I_t\}$ contains observed image at the t frame and those before t frame. The term $p(X_t/X_{t-1})$ is called dynamical model, and usually modeled by a Gaussian distribution in which each parameter of X_t is treat independent. And $p(I_t/X_t)$ is called observation model, which is a probability to describe the target tracked.

As the integration in Eq. (1) is intractable analytically due to the non-Gaussian form of $p(I_t/X_t)$, we resort to particle filtering-based sampling. The particle is represented by the pixel-wise spatial pyramid.

3.2 Pixel-Wise Spatial Pyramid

The image I is represented as a two-dimensional lattice of a one-dimensional intensity image or a three-dimensional color image. Let $F(x, y)$ be the d -dimensional appearance vector extracted from I at the spatial location (x, y)

$$F(x, y) = \Gamma(I, x, y) \quad (2)$$

where Γ can be any mapping such as color, intensity, image gradient I_x, I_{xx}, \dots , edge, texture etc. The original pixel feature vector in [13] includes spatial attributes that are obtained from pixel coordinate values, but we only use the d -dimensional appearance vector and the spatial layout is exploited through the spatial pyramid. So a $M \times N$ rectangular region R forms a two-dimensional matrix of pixel feature vectors W ,

$$W_{(M*N) \times d} = [F_1, F_2, \dots, F_{M*N}] \quad (3)$$

Here, we employ the intensity and texture information to generate the five-dimensional individual pixel vector $F(x, y)$,

$$F(x, y) = [I(x, y), |I_x(x, y)|, |I_y(x, y)|, |I_{xx}(x, y)|, |I_{yy}(x, y)|] \quad (4)$$

In order to introduce spatial information, we follow the scheme proposed by Lazebnik *et al.* which is based on pyramid matching [20]. Spatial pyramid matching is a simple yet effective approach to compare similarity between images. The image is divided into a sequence of increasingly finer spatial grids by repeatedly doubling the number of divisions in each axis direction. Let X and Y be two sets of vectors in a d -dimensional feature space which are obtained from two images. In general, SIFT vectors are used, but here they are pixel feature vectors. Pyramid matching is implemented by taking a weighted sum of the number of matches that occur at each level of resolution. Supposing we have constructed a sequence of resolutions $0, 1, \dots, L$, then we have 2^l sub-regions for the l th resolution. Let H_X^l and H_Y^l denote the histograms of X and Y at resolution l , so the histogram intersection can be computed as $I(H_X^l, H_Y^l) = \sum_{i=1}^{2^l} \min(H_X^l(i), H_Y^l(i))$. Then the overall similarity between X and Y is defined as

$$S(H_X, H_Y) = \sum_{l=0}^L w_l I(H_X^l, H_Y^l) \quad (5)$$

where the weight is $w_l = \max(\frac{1}{2^l}, \frac{1}{2^{L-l+1}})$. The details can be found in [19]. In our case, the set of pixel feature vectors are quantized by K-means with a codebook size 25, and the number of levels is limited to $L = 2$ to prevent over fitting.

3.3 Update Scheme

Though the trackers such as [1, 5] without update perform well under some circumstances, the update of tracking model is essential to cope with appearance changes. The template tracking methods is often updated by the approach based on matching score. For example, the template update mechanism in [2] is defined as

$$q^{i+1} = \alpha \pi q^i + (1 - \alpha)(1 - \pi)p(y_i) \quad (6)$$

where $\alpha = 0.85$ is a weighting factor to control the speed of the updates, q^i is the template at frame i , and $\pi = \rho[p(y_i), q]$ is the Bhattacharyya coefficient between the current template and the optimal candidate found in the i^{th} frame. The rule indicates that the update of the template will become minimal, if the template and the optimal candidate are well-matched. However, this method is not suitable in our tracker, because both the codebook and the histogram representation need to be updated. On the other hand, since there is no weak learner in our tracker, our appearance model also cannot evolve like the methods in [6,7].

We develop a distance-based scheme to update the codebook and hybrid feature map to obtain the target appearance model, as depicted in Fig. 3.

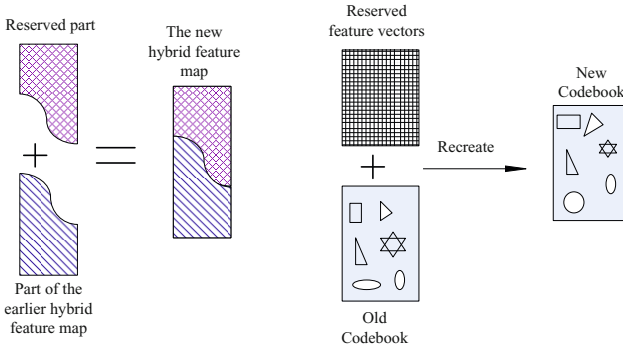


Fig. 3. Update Scheme

Each pixel feature vector in the current frame has a minimal distance to the code words. Note that the new pixel feature vectors those are nearer to the previous code words in the current frame are more likely to be the target elements. All the current pixel feature vectors are sorted in a queue ascendingly according to the minimal distances. During the process of update, for the adaptive module, an empirical threshold is used to keep the valuable information and remove the noise, particularly part of the occlusion sector. The top 15 percent in the queue is captured as the reserved feature vectors for the soft stable module. The current valuable information, i.e., part of the pixel feature vectors, together with the previous cluster centers (code words) are employed to generate a new codebook.

Since the spatial pyramid matching focuses on the matching between images, we propose a hybrid feature map according to the spatial layout to keep the cumulated instrumental information. The pixel feature vectors currently reserved are part of the hybrid feature map, and the remainders are the opposite spatial part in the earlier hybrid feature map. Figure 4 shows some of the hybrid maps in Girl and Shaking sequences (f denotes the frame number). The left shows the object region, the center depicts the hybrid maps of the adaptive module, and the right displays the hybrid maps of the soft stable module. It can be found that the hybrid maps of the adaptive module catch more appearance changes than the maps of soft stable module do, because of the different update rates of them.

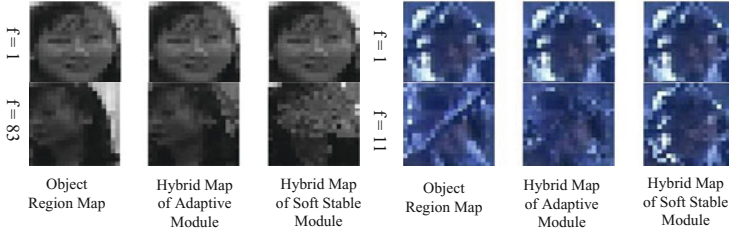


Fig. 4. Hybrid Maps of Girl and Shaking

4 Biased Multiplicative Formula

In order to take advantage of the reference information, the soft reference model, as well as online updated appearance model at the same time, we use biased multiplicative formula to fuse them. Suppose that the similarity metric matching scores are S_r, S_{sr}, S_o , i.e., likelihood scores between the candidates and the above three models. The fusion equation is defined as

$$S_f = S_r * S_{sr} * S_o \quad (7)$$

Firstly, this can be interpreted that the (soft) reference model is used to verify the judgment of the online appearance model. Furthermore, this fusion scheme can find the balance key between the three tracking modules. Finally, to make the tracker less prone to drift, we choose the candidate which has a bigger S_r or $S_r * S_{sr}$.

5 Implementation and Experiments

During the experiments, we compare our algorithm to current state-of-the-art methods, i.e., IVT, FragTrack and MILTrack, on publicly available datasets. Babenko *et al.* showed superior results comparing their method (MILTrack) to On-line Boosting and FragTrack. IVT is a successful online learning algorithm using incremental subspace representation. FragTrack benefitting from the division-combination patches scheme is robust to occlusion and perform well on several challenging sequences.

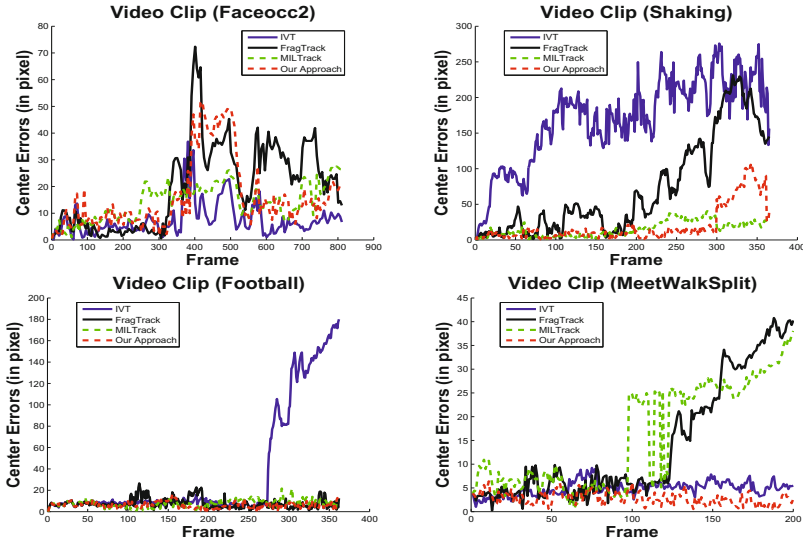
Throughout the experiments, we use seven challenging video sequences regarding e.g. moving cameras, occlusions, background clutters, 3-D motion and illumination changes. The ground truth for sequences: Girl and Faceocc2 are from [8]. ShopAssistant2cor and MeetWalkSplit come from the CAVIAR database. Shaking, Football and Skating1(low frame rate) are from [21].

5.1 Quantitative Evaluation

In this experiment, we would like to benchmark our method on the following sequences: Faceocc2, Shaking, Football and MeetWalkSplit. Table 1 and Fig. 5 depict the results based on the mean pixel error: our method yields the best

Table 1. Average center location errors in pixels

Sequence	IVT	FragTrack	MILTrack	Our approach
Faceocc2	7.5	19.9	14.3	16.1
Shaking	170.4	70.6	15.1	19.8
Football	37.3	7.5	7.5	6.1
MeetWalkSplit	4.9	14.2	16.0	2.8

**Fig. 5.** Error curves of some testing video sequences

scores in two sequences: MeetWalkSplit and Football, and gains the second best results in the sequence: Shaking. IVT performs best in Faceocc2 seq., but fails in Shaking and Football sequences, due to the severe illumination variation, out of plane rotation or viewpoint changes. FragTrack fails in Shaking seq. because of the drastic illumination changes. Though our tracker locates the object accurately in the first part of Shaking seq., it drifts in the tail because there is a combination of pose change and drastic illumination change. Our tracker even loses the target in the middle of Faceocc2 seq. because of the severe occlusion, but then recovers due to the utilization of stable and soft stable modules.

5.2 Performance of the Individual Tracking Module

Here we investigate the behavior of our three appearance modules respectively on two sequences, Girl and Shaking. The average pixel error is given in Fig. 6. The reference module works well when the appearance of the object is close to it. The soft reference module and adaptive module seem to perform poor in

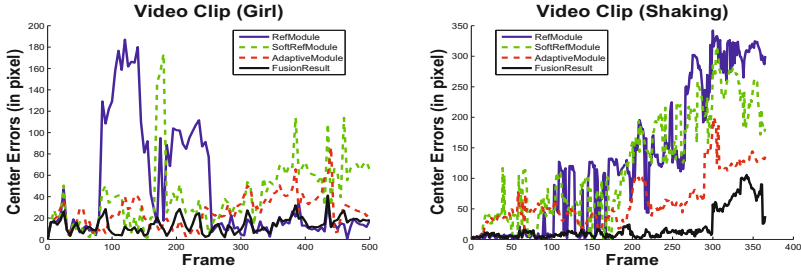


Fig. 6. Evaluation of the separate modules and fusion tracker

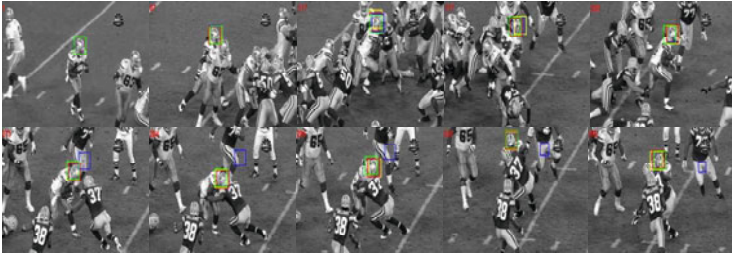


Fig. 7. Comparison between IVT, FragTrack, MILTrack and our method in Football

the sequences. But through the biased multiplicative formula, the fusion result becomes much more accurate and robust. Unfortunately, in the later part of sequence Shaking, all the results of three modules are far from the ground truth, leading drifting of the final output.

5.3 Qualitative Evaluation

We evaluate the performance of our tracking method through comparing with IVT, FragTrack and MILTrack. The bounding boxes for target of IVT, FragTrack, MILTrack and our approach are blue, yellow, green and red respectively.

Background clutter. In Fig. 7, we test Football seq. that includes severe background clutter, of which appearance is similar to that of the target. In the case of IVT, the bounding box drifts when two players collided with each other and cannot recover as illustrated in the second row of Fig. 7. Our method, FragTrack and MILTrack overcome this problem, and our approach is more accurate than them.

Occlusion. Figure 8 shows the tracking results for the pedestrian in ShopAssistant2cor. While the person of ShopAssistant2cor is severely occluded in the frames from 185 to 220, all the methods successfully track the target. But after that, the three other trackers only locate part of the object. Note that there



Fig. 8. Comparison between IVT, FragTrack, MILTrack and our method in ShopAssistant2cor

are also persons with similar color distribution comparing to the object. Our approach never drifts throughout the sequence.

3-D motion and moving camera. We present the tracking results of Girl in Fig. 9. Since our method and IVT employ the affine motion model, so we can find more accurate location of the object. Note that, in order to evaluate the motion model, the initial bounding boxes in IVT and our approach are made to be a little smaller than the boxes in MILTrack and FragTrack. The results demonstrate that IVT can track the girl in the first few frames, but fails after the girl rotates. Our method, FragTrack and MILTrack can locate the girl, but FragTrack fails during the frames from 20 to 60, MILTrack drifts in the end.

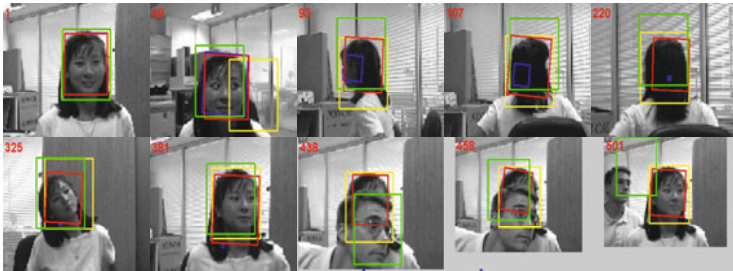


Fig. 9. Comparison between IVT, FragTrack, MILTrack and our method in Girl

Illumination change and pose variations. We assess the capability of the tracking methods regarding illumination change and drastic pose variations in Skating1(low frame rate). As shown in Fig. 10, our method covers these challenges, while IVT drifts after a few frames, the other two methods locate part of object during some of the frames. Note that there are also abrupt motion and occlusion in this sequence. For example, abrupt motion and serious occlusion can be found in frame 75 and 141 respectively.

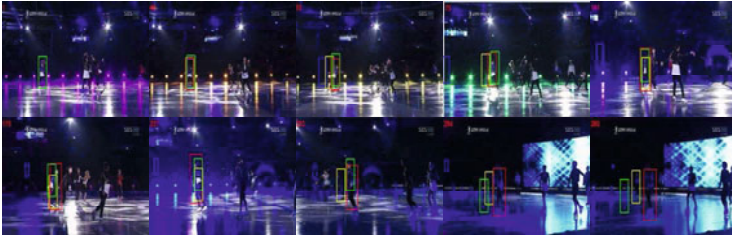


Fig. 10. Comparison between IVT, FragTrack, MILTrack and our method in Skating1(low frame rate)

6 Conclusion and Future Work

In this paper, we present a new algorithm to track object whose appearance changes drastically. We fuse three tracking modules with different update rate through biased multiplicative formula to achieve the balance between robustness and adaptivity of the tracker. Particularly, the pixel-wise spatial pyramid including several appearance features and spatial layout and the hybrid feature map accommodating the cumulated valuable pixel feature vectors play the crucial role during tracking process. We demonstrate comparative performance with the state-of-the-art tracking methods in sequences of challenging circumstances.

Future work: Through the experiments, we find that the K-means clustering is not strong enough to quantize the pixel feature vectors, because the dimension of the vector is high or the different information locates in different feature space. Other quantification methods, e.g. Gaussian Mixture Model (GMM) or Histogram Intersection Kernel (HIK), could be used to create a better codebook.

Acknowledgement. The work was supported by the Fundamental Research Funds for the Central Universities, No. DUT10JS05, and the National Natural Science Foundation of China (NSFC), No.61071209.

References

1. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: CVPR, vol. 2, pp. 142–149 (2000)
2. Cannons, K., Wildes, R.: Spatiotemporal oriented energy features for visual tracking. In: Yagi, Y., Kang, S.B., Kweon, I.S., Zha, H. (eds.) ACCV 2007, Part I. LNCS, vol. 4843, pp. 532–543. Springer, Heidelberg (2007)
3. Wang, H., Suter, D., Schindler, K.: Effective appearance model and similarity measure for particle filtering and visual tracking. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3953, pp. 606–618. Springer, Heidelberg (2006)
4. Ross, D.A., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking. *Int. J. Comput. Vision* 77(1), 125–141 (2008)
5. Adam, A., Rivlin, E., Shimshoni, I.: Robust fragments-based tracking using the integral histogram. In: CVPR, pp. 798–805 (2006)

6. Grabner, H., Grabner, M., Bischof, H.: Real-time tracking via on-line boosting. In: BMVC, vol. 1, pp. 47–56 (2006)
7. Avidan, S.: Ensemble tracking. In: CVPR, vol. 2, pp. 494–501 (2005)
8. Babenko, B., Yang, M.H., Belongie, S.: Visual tracking with online multiple instance learning. In: CVPR, pp. 983–990 (2009)
9. Grossberg, S.: Competitive learning: From interactive activation to adaptive resonance. In: NNNI, pp. 213–250 (1998)
10. Grabner, H., Leistner, C., Bischof, H.: Semi-supervised on-line boosting for robust tracking. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 234–247. Springer, Heidelberg (2008)
11. Stalder, S., Grabner, H., Gool, L.V.: Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition. In: ICCV (2009)
12. Jakob, S., Christian, L., Amir, S., Thomas, P.: Prost: Parallel robust online simple tracking. In: CVPR (2010)
13. Porikli, F., Tuzel, O., Meer, P.: Covariance tracking using model update based on lie algebra. In: CVPR, pp. 728–735 (2006)
14. Arif, O., Vela, P.: Non-rigid object localization and segmentation using eigenspace representation. In: ICCV (2009)
15. Lampert, C.H., Blaschko, M.B., Hofmann, T.: Beyond sliding windows: Object localization by efficient subwindow search. In: CVPR, pp. 1–8 (2008)
16. Nowak, E., Jurie, F., Triggs, B.: Sampling strategies for bag-of-features image classification. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 490–503. Springer, Heidelberg (2006)
17. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR, pp. 2169–2178 (2006)
18. Lowe, D.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision* 60(2), 91–110 (2004)
19. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, vol. 1, pp. 886–893 (2005)
20. Grauman, K., Darrell, T.: The pyramid match kernel: Discriminative classification with sets of image features. In: ICCV, vol. 2, pp. 1458–1465 (2005)
21. Kwon, J., Lee, K.M.: Visual tracking decomposition. In: CVPR (2010)

Compressive Evaluation in Human Motion Tracking

Yifan Lu¹, Lei Wang¹, Richard Hartley^{1,3}, Hongdong Li^{1,3}, and Dan Xu²

¹ School of Engineering, CECS, Australian National University

² Department of Computer Science and Engineering, SISE, Yunan University

³ Canberra Research Labs, National ICT Australia

{Yifan.Lu,Lei.Wang,Richard.Hartley,Hongdong.Li}@anu.edu.au,
danxu@ynu.edu.cn

Abstract. The powerful theory of compressive sensing enables an efficient way to recover sparse or compressible signals from non-adaptive, sub-Nyquist-rate linear measurements. In particular, it has been shown that random projections can well approximate an isometry, provided that the number of linear measurements is no less than twice of the sparsity level of the signal. Inspired by these, we propose a compressive anneal particle filter to exploit sparsity existing in image-based human motion tracking. Instead of performing full signal recovery, we evaluate the observation likelihood directly in the compressive domain of the observed images. Moreover, we introduce a progressive multilevel wavelet decomposition staged at each anneal layer to accelerate the compressive evaluation in a coarse-to-fine fashion. The experiments with the benchmark dataset HumanEvaII show that the tracking process can be significantly accelerated, and the tracking accuracy is well maintained and comparable to the method using original image observations.

1 Introduction

Compressive sensing (CS) acquires and reconstructs compressible signals from a small number of non-adaptive linear random measurements by combining the steps of sampling and compression [1,2,3,4]. It enables the design of new kinds of compressive imaging systems, including a single pixel camera [5] with some attractive features, including simplicity, low power consumption, universality, robustness, and scalability. Recently, there has been a growing interest of compressive sensing in computer vision and it has been successfully applied to face recognition, background subtraction, object tracking and other problems. Wright et al [6] represented the test face image in a linear combination of training face images. Their representation is naturally sparse, involving only a small fraction of the overall training database. Such a problem of classifying among multiple linear regression models can be then solved efficiently via L_1 -minimisation which seeks the sparsest representation and automatically discriminates between the various classes presented in the training set. Cevher et al [7] cast the background subtraction problem as a sparse signal recovery problem and solved by greedy

methods as well as total variation minimisation as convex objectives to process field data. They also showed that it is possible to recover the silhouettes of foreground objects by learning a low-dimensional compressed representation of the background image without learning the background itself to sense the innovations or the foreground objects. Mei et al [8] formulated the tracking problem similar to [6]. In order to find the tracking target at a new frame, each target candidate is sparsely represented in the space spanned by target templates and trivial templates. The sparse representation is obtained by solving an $L1$ -regularised least squares problem to find good target templates. Then the candidate with the smallest projection error is taken as the tracking target. Subsequent tracking is continued using a Bayesian state inference framework in which a particle filter is used for propagating sample distributions over time.

Unlike above works, many data acquisition/processing applications do not require obtaining a precise reconstruction, but rather are only interested in making some kind of evaluations on the objective function. Particularly, human motion tracking essentially attempts to find the optimal value of the observation likelihood function. Therefore, we propose a new framework, called Compressive Annealed Particle Filter, for such a situation that bypasses the reconstruction and performs evaluations solely on compressive measurements. It has been proven [1] that the random projections can approximately preserve an isometry and pairwise distance, when the number of the linear measurements is large enough (still much smaller than the original dimension of the signal). Moreover, noticing the annealing schedule is a coarse-to-fine process, we introduce the staged wavelet decomposition with respect to each anneal layer so that the increasing anneal variable is absorbed into the wavelet decomposition. As a result, the number of compressive measurements is progressively increased to gain computational efficiency.

The rest of the paper is organised as follows. Section 2 describes the human body template. In Section 3, we provide a brief overview of the theoretical foundation of Compressive Sensing, followed by Compressive Annealed Particle Filter in Section 4 and the results of experiments with the HumanEvaII dataset in Section 5. Finally, Section 6 concludes with a brief discussion of our results and directions for future work.

2 Human Body Template

The textured body template in our work uses a standard articulated-joint parametrisation to describe the human pose, further leading to an effective representation of the human motion over time. Our articulated skeleton consists of 10 segments and is parameterised by 25 degrees of freedom (DOF) in Figure 1. It is registered to a properly scaled template skin mesh by Skeletal Subspace Deformation (SSD) [9]. Then, shape details and texture are recovered by an interactive volumetric reconstruction and the texture registration procedure. At last, the template model is imported to commercial software to be finalised according to the real subject. The example of the final template model is illustrated in Figure 1.

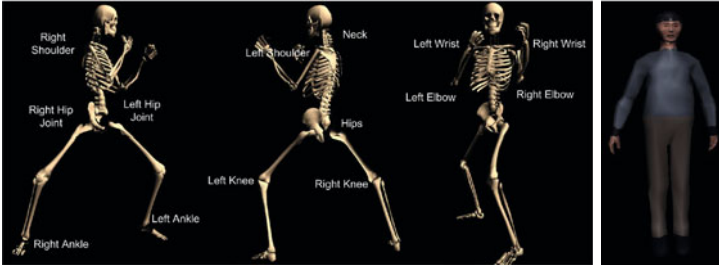


Fig. 1. From left to right: the articulated skeleton parameterised by 25 DOF and the textured template model after manual refinements used in this work

3 Compressive Sensing

The novel theory of Compressive Sensing (CS) [1,2,3,4] provides a fundamentally new approach to data acquisition that provides a better sampling and compression when the underlying signal is known to be sparse or compressible, yielding a sub-Nyquist sampling criterion.

3.1 Signal Sparse Representation

We consider that a signal $\mathbf{f} \in \mathbb{R}^N$ is sparse in some orthonormal basis $\Psi \in \mathbb{R}^{N \times N}$ and can be represented as $\mathbf{f} = \Psi \mathbf{f}'$. If there are only a few significant entries in \mathbf{f}' , and insignificant entries can be discarded without much loss, then \mathbf{f}' can be well approximated by \mathbf{f}'_K that is constructed by keeping the K largest entries of \mathbf{f}' unchanged and setting all remaining $N - K$ entries to zero. Then $\mathbf{f}_K = \Psi \mathbf{f}'_K$ is so called K -sparse representation. Since Ψ is an orthonormal matrix, hence $\|\mathbf{f} - \mathbf{f}_K\|_2 = \|\mathbf{f}' - \mathbf{f}'_K\|_2$. If \mathbf{f}' is sparse or compressible in the sense that the sorted magnitudes of its components x_i decay quickly, then the relative error $\frac{\|\mathbf{f} - \mathbf{f}_K\|_2}{\|\mathbf{f}\|_2}$ is also small. Therefore, the perceptual loss of \mathbf{f}_K with respect to \mathbf{f} is hardly noticeable.

3.2 L1 Minimisation Recovery

Compressive sensing nevertheless surprisingly predicts that reconstruction from vastly undersampled non-adaptive measurements is possible—even by using efficient recovery algorithms. Let us consider M ($M \ll N$) non-adaptive linear measurements \mathbf{z} (so called *Compressive Measurement*) of a signal \mathbf{f} using $\mathbf{z} = \Phi \mathbf{f}$, where $\Phi \in \mathbb{R}^{M \times N}$ denotes the measurement matrix. Since $M \ll N$, the recovery of \mathbf{f} from \mathbf{z} is underdetermined. If, however, the additional assumption is imposed that the vector \mathbf{f} has sparse representation, then the recovery can be realised by searching for the sparsest vector \mathbf{f}^* that is consistent with the measurement vector $\mathbf{z} = \Phi \Psi \mathbf{f}'$. The finest recovery $\mathbf{f}^* = \Psi \mathbf{f}'^*$ is achieved when the sparsest vector \mathbf{f}'^* is found. This leads to solving a L_0 -minimisation problem.

Unfortunately, the combinatorial L_0 -minimisation problem is NP hard in general [10]. In [2] Candes et al have shown that the L_1 norm yields the equivalent solution to the L_0 norm, resulting in solving an easier linear program, for which efficient solution methods already exist. When the measurement process involves a small stochastic error term $\|\eta\|_2 \leq \epsilon$, $\mathbf{z} = \Phi\Psi\mathbf{f}' + \eta$, the L_1 -minimisation approach considers the solution of:

$$\min \|\mathbf{f}'\|_1 \quad \text{subject to} \quad \|\Phi\Psi\mathbf{f}' - \mathbf{z}\|_2 \leq \epsilon \quad (1)$$

This is an instance of second order cone programming [3] which has a unique convex solution.

The exact recovery from non-adaptive linear measure is not universal but conditional. The primary result [4] of CS states, if Φ is incoherent with Ψ so that the coherence $\mu(\Phi, \Psi) = \sqrt{N} \max_{l,k \in [1,N]} |\langle \phi_l, \psi_k \rangle|$ [1] is close to 1 and $M \geq C\mu^2(\Phi, \Psi)K \log N/\sigma$ for some positive constant C and small values of σ , then \mathbf{f}' in $\mathbf{z} = \Phi\mathbf{f} = \Phi\Psi\mathbf{f}'$ can be exactly recovered with overwhelming probability $1 - \sigma$. Moreover, it turns out that a randomly generated matrix Φ from an isotropic sub-Gaussian distribution (e.g. from i.i.d. Gaussian or Bernoulli/ Rademacher 1 vectors) is incoherent with high probability to an arbitrarily fixed basis Ψ .

4 Compressive Annealed Particle Filtering

The proposed approach resides on the APF framework that is first introduced in human tracking by Deutscher et al. [11]. APF incorporates simulated annealing [12] for minimising an energy function $E(\mathbf{y}_t, \mathbf{x}_t)$ or, equivalently, maximising the observation likelihood $p(\mathbf{y}_t|\mathbf{x}_t)$ that measures how well a particle (an estimate pose configuration) \mathbf{x}_t fits the observation \mathbf{y}_t at time t . The observation likelihood is essential for APF in order to approximate the posteriori distribution, and it is often formulated in a modified form of the Boltzmann distribution:

$$p(\mathbf{y}_t|\mathbf{x}_t) = \exp\{-\lambda E(\mathbf{y}_t, \mathbf{x}_t)\} \quad (2)$$

where the annealing variable λ is $1/(k_B T_t)$, an inverse of the product of the Boltzmann constant k_B and the temperature T_t at time t . The optimisation of APF is iteratively done according to a predefined L -phase schedule $\{\lambda = \lambda_1, \dots, \lambda_L\}$, where $\lambda_1 < \lambda_2 < \dots < \lambda_L$, known as the annealing schedule. At time t , considering a single phase l , initial particles are outcomes from the previous phase $l - 1$ or drawn from the temporal model $p(\mathbf{x}_t|\mathbf{x}_{t-1})$. Then, all particles are weighted by their observation likelihood $p(\mathbf{y}_t|\mathbf{x}_t)$ and resampled probabilistically to select good particles which are highly likely to near the global optimum. Finally, particles are perturbed by a Gaussian noise with a diagonal covariance matrix P_l [2].

¹ ϕ_l is a row of Φ . ψ_k is a column of Ψ . To simplify the notation, ϕ_l can be concatenated as the basis with N elements so that $\langle \phi_l, \psi_k \rangle$ is always computable.

² The perturbation covariance matrix P_l is used to adjust the search range of particles.

Considering the pose space model in a dynamic structure that consists of a sequence of estimate poses \mathbf{x}_t at successive time $t = 1, 2, \dots$, and each pose is associated with an image observation \mathbf{y}_t^{obs} or a compressive measurement \mathbf{z}_t^d . At time t , the compressive measurement can be defined by:

$$\begin{aligned}\mathbf{z}_t^d &= \Phi \Psi \mathbf{y}_t^d \\ &= \Phi \Psi (\mathbf{y}_t^{obs} - \mathbf{y}_t^{bg}) \\ &= \mathbf{z}_t^{obs} - \mathbf{z}_t^{bg}\end{aligned}\quad (3)$$

where, Ψ denotes wavelet basis. In particular, \mathbf{y}_t^d is the difference image generated by subtracting the background image \mathbf{y}_t^{bg} from the original observation image \mathbf{y}_t^{obs} . It is known that the images acquired from the natural scene have highly sparse representation in the wavelet domain. The difference image calculated by subtracting the static background from the observation image has more pixel values close to zero, hence, the difference image $\Psi \mathbf{y}_t^d$ is also highly sparse and compressible in general.

On the other hand, given the estimate state \mathbf{x}_t , the estimate compressive measurement $\hat{\mathbf{z}}_t^d$ of the difference image can be calculated by subtracting the background image \mathbf{y}_t^{bg} from the synthetic foreground image $s^{fg}(\mathbf{x}_t)$, which is generated by projecting the human model with the pose \mathbf{x}_t and camera parameters onto the image plane. This difference image is also compressible in the wavelet domain so that it can be defined by:

$$\begin{aligned}\hat{y}_{t,i}^d &= sil_i(\mathbf{x}_t) * (s_i^{fg}(\mathbf{x}_t) - y_{t,i}^{bg}) \quad i = 1, \dots, N \\ \hat{\mathbf{z}}_t^d &= \Phi \Psi \hat{\mathbf{y}}_t^d\end{aligned}\quad (4)$$

where, $sil(\mathbf{x}_t)$ is a synthetic silhouette mask generated by the estimate state \mathbf{x}_t which has 0s on all background entries and 1s on all the foreground entries. This mask operation is used to make the synthetic difference image is comparable to the original difference image.

4.1 Restricted Isometry Property and Pairwise Distance Preservation

Another important result of CS is the Restricted Isometry Property (RIP) [1] which characterises the stability of nearly orthonormal measurement matrices. A matrix Φ satisfies RIP of order K if there exists an isometry constant $\sigma_K \in (0, 1)$ as the smallest number, such that $(1 - \sigma_K) \|\mathbf{f}'\|_2^2 \leq \|\Phi \mathbf{f}'\|_2^2 \leq (1 + \sigma_K) \|\mathbf{f}'\|_2^2$ holds for all $\mathbf{f}' \in \Sigma_K = \{\mathbf{f}' \in \mathbb{R}^N : \|\mathbf{f}'\|_0 \leq K\}$. In other words, Φ is an approximate isometry for signals restricted to be K -sparse and approximately preserves the Euclidean length, interior angles and inner products between the K -sparse signals. This reveals the reason why CS recovery is possible because Φ embeds the sparse signal set Σ_K in \mathbb{R}^M while no two sparse signals in \mathbb{R}^N are mapped to the same point in \mathbb{R}^M .

If Φ has i.i.d. Gaussian entries and $M \geq 2K$, then there always exists $\sigma_{2K} \in (0, 1)$ such that all pair-wise distances between K -sparse signals are well preserved [13]:

$$(1 - \sigma_{2K}) \leq \frac{\|\Phi \mathbf{f}'_i - \Phi \mathbf{f}'_j\|_2^2}{\|\mathbf{f}'_i - \mathbf{f}'_j\|_2^2} \leq (1 + \sigma_{2K}). \quad (5)$$

Meanwhile, Baraniuk and Wakin [14] present a Johnson-Lindenstrauss (JL) lemma [15] formulation with the stable embedding of a finite point cloud under a random orthogonal projection, which has a tighter lower bound for M .

Lemma 1. [14] *Let \mathbb{Q} be a finite collection of points in \mathbb{R}^N . Fix $0 < \sigma < 1$ and $\beta > 0$. Let $\Phi \in \mathbb{R}^{M \times N}$ be a random orthogonal matrix and*

$$M \geq \left(\frac{4 + 2\beta}{\sigma^2/2 + \sigma^3/3} \right) \ln(\#\mathbb{Q})$$

If $M \leq N$, then, with probability exceeding $1 - (\#\mathbb{Q})^{-\beta}$, the following statement holds: For every $\mathbf{f}'_i, \mathbf{f}'_j \in \mathbb{Q}$ and $i \neq j$

$$(1 - \sigma) \sqrt{\frac{M}{N}} \leq \frac{\|\Phi \mathbf{f}'_i - \Phi \mathbf{f}'_j\|_2}{\|\mathbf{f}'_i - \mathbf{f}'_j\|_2} \leq (1 + \sigma) \sqrt{\frac{M}{N}}$$

where a random orthogonal matrix can be constructed by performing the Householder transformation [16] on M random length- N vectors having i.i.d. Gaussian entries, assuming the vectors are linearly independent.

4.2 Multilevel Wavelet Likelihood Evaluation on Compressive Measurements

The above Equation (5), Lemma (1) and orthonormality of Ψ guarantee the pairwise distance to be approximately preserved provided that M is sufficient large. Therefore the CS recovery is not necessary to evaluate the observation likelihood. Instead, the observation likelihood can be directly calculated via the distance of compressive measurements in Equation (3) and (4).

$$p(\mathbf{y}_t | \mathbf{x}_t) = \exp\{-\lambda \|\mathbf{z}_t^d - \hat{\mathbf{z}}_t^d\|_2\} \quad (6)$$

Notice $\lambda > 0$, the above equation can be transformed as:

$$\begin{aligned} p(\mathbf{y}_t | \mathbf{x}_t) &= \exp\{-\|\lambda \mathbf{z}_t^d - \lambda \hat{\mathbf{z}}_t^d\|_2\} \\ &= \exp\{-\|\Phi \lambda (\Psi \mathbf{y}_t^d - \Psi \hat{\mathbf{y}}_t^d)\|_2\} \end{aligned} \quad (7)$$

In the equation (7), $\Psi \mathbf{y}_t^d$ and $\Psi \hat{\mathbf{y}}_t^d$ are wavelet coefficients. According to multilevel wavelet decomposition, we construct two wavelet coefficient sequences of $\mathbf{C} = \{\mathbf{c}_i | i = 1, 2, \dots\}$ and $\hat{\mathbf{C}} = \{\hat{\mathbf{c}}_i | i = 1, 2, \dots\}$ for $\Psi \mathbf{y}_t^d$ and $\Psi \hat{\mathbf{y}}_t^d$. Furthermore, $\mathbf{c}_i \subset \mathbf{c}_{i+1}$ the current level wavelet coefficient are always a subset of its super level wavelet coefficient. Hence, $\|\mathbf{c}_i\|_1 < \|\mathbf{c}_{i+1}\|_1$ and \mathbf{C} is considered a monotonically increasing sequence in terms of the magnitude (the same can be applied to $\hat{\mathbf{C}}$). For instance, a four-level wavelet coefficient sequence is

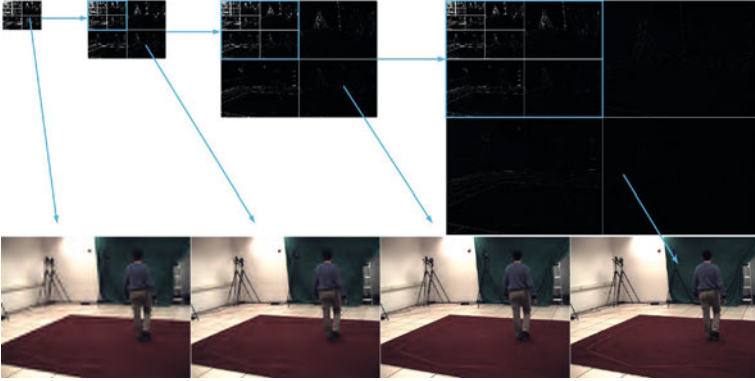


Fig. 2. The number of wavelet coefficients is progressively elevated as the wavelet decomposition process so that details are gradually enhanced through the anneal schedule. From left to right, we show 4 levels wavelet decomposition coefficients at the top of the figure. 1) using only the $K_4 = 2805$ largest coefficients (about 18.39% over all the level 4 coefficients) at the level 4, 2) $K_3 = 4345$ (7.18%) at the level 3, 3) $K_2 = 12086$ (5.01%) at the level 2 and 4) $K_1 = 30000$ (3.11%) at the level 1. The observation images at the bottom are reconstructed by using corresponding K_g sparse wavelet coefficients.

shown in the top of Figure 2. Obviously, $\mathbf{C}^\Delta = \mathbf{C} - \hat{\mathbf{C}}$ has the same monotonically increasing property $\|\mathbf{c}_i^\Delta\|_1 < \|\mathbf{c}_{i+1}^\Delta\|_1$. If defining a series of variables $\lambda_i = \|\mathbf{c}_{i+1}^\Delta\|_1 / \|\mathbf{c}_i^\Delta\|_1$ $i = 1, 2, \dots$, where $\lambda_i < \lambda_{i+1}$, alternatively, this monotonically increasing sequence \mathbf{C}^Δ can be described by $\mathbf{C}^\Delta = \{\mathbf{c}_1^\Delta, \lambda_1 \mathbf{c}_1^\Delta, \lambda_2 \mathbf{c}_1^\Delta, \dots\}$. In other words, we always can construct a monotonically increasing wavelet coefficient sequence \mathbf{C}^Δ that has an equivalent counterpart series of λ . The precise value of λ for each anneal layer is not very critical, since λ is only used to roughly control the optimisation convergence rate. Therefore, we design directly evaluating the coarse-to-fine wavelet coefficients in difference levels to simulate increasing λ_l at each layer l . Then, an alternative of Equation (7) is given by:

$$p(\mathbf{y}_t | \mathbf{x}_t) = \exp\{-\|\Phi(l)(\Psi(l, \mathbf{y}_t^d) - \Psi(l, \hat{\mathbf{y}}_t^d))\|_2\} \quad (8)$$

where, $\Psi(l, \mathbf{y}_t^d)$ is wavelet coefficients of \mathbf{y}_t^d at the l layer associated to the level g decomposition, and it has N_l wavelet coefficients. With l is increasing, g is decreasing and the more details encoded in wavelet coefficients $\Psi(l, \mathbf{y}_t^d)$ are used. For instance, as shown in Figure 2, $\Phi(l)$ is a $M_l \times N_l$ sub-matrix of Φ . $M_l = 2K_g$ is determined according to the sparsity K_g of the g level wavelet coefficients.

5 Experiments

Experiments are conducted on the benchmark dataset HumanEvaII [17] that contains two 1260-frame image sequences from 4 colour calibrated cameras synchronised with Mocap data at 60Hz. Those tracking subjects perform three different actions including walking, jogging and balancing. To generate compressive

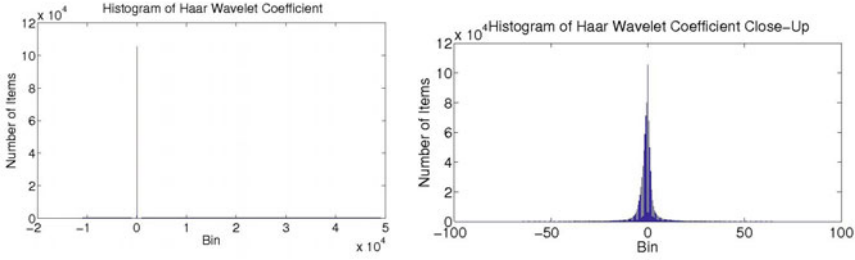


Fig. 3. Wavelet Coefficient Histogram and Wavelet Coefficient Histogram (close-up view) showing that 95% coefficients have very small values close to zero

measurements, we apply the 8-level haar wavelet 2D decomposition [18] to all observation images. The wavelet coefficients appear highly sparse, most of which are close to zero as illustrated in Figure 3. For instance, using solely the 30000 largest wavelet coefficients we are able to reconstruct the 964320 colour components of 656×490 RGB image with hardly noticeable perceptual loss. For the multilevel evaluation (Equation 8), the four sparsity levels $K_1 = 30000$, $K_2 = 12086$, $K_3 = 4345$ and $K_4 = 2805$ are evenly allocated in the 10 anneal layers³. The $M_l = 2K_g$ rows of Φ are drawn i.i.d. from the normal distribution $N(0, 1/M_l)$ to approximately preserve the isometry as shown in Equation (5). On the other hand, the single level evaluation Equation (6) is used with a tight lower bound for M shown in Lemma (II). We presume there are one observation image and maximum 2000⁴ synthetic images generated in the evaluation for each view and each frame. Then, for the 1260-frame sequence, there are total 2521260 unique compressive measurements required for tracking. Let $\sigma = 0.1$, $\beta = 1$ and $\#\mathbb{Q} = 2521260$, so $M = \left(\frac{4+2\beta}{\sigma^2/2+\sigma^3/3}\right) \ln(\#\mathbb{Q}) = 16583$. Moreover, the M rows of the Φ are constructed by drawing i.i.d. entries from the normal distribution $N(0, 1/M)$ and performing the Householder transformation to orthogonalise Φ . Therefore, with high probability $1 - 1/2521260$, Φ approximately preserves the pairwise distance. we also verified the performance of the number of compressive measurements in cases of $M = 10000$ and $M = 5000$.

As illustrated in the experimental results of HumanEvaII Subject 2 (the top of Figure 4), the evaluation using original images as the evaluation input obtains $54.5837 \pm 4.7516mm$ ⁵. The multilevel evaluation achieves the stable results $56.9442 \pm 4.4581mm$ which is comparable with the results using original images. When using the single level evaluation with $M = 16583$ compressive measurements, the tracking performance appears poorer than the multilevel evaluation but still maintains within $65.7548 \pm 5.4351mm$. When the number of compressive measurements are further reduced to $M = 10000$ and $M = 5000$, the

³ Using $M_1 = 2 \times 2805$, $M_2 = 2 \times 2805$, $M_3 = 2 \times 2805$, $M_4 = 2 \times 4345$, $M_5 = 2 \times 4345$, $M_6 = 2 \times 4345$, $M_7 = 2 \times 12086$, $M_8 = 2 \times 12086$, $M_9 = 2 \times 30000$, $M_{10} = 2 \times 30000$.

⁴ Given 10 layers and 200 particles as the maximum.

⁵ The results are statistically presented by mean \pm standard deviation in Millimetres.

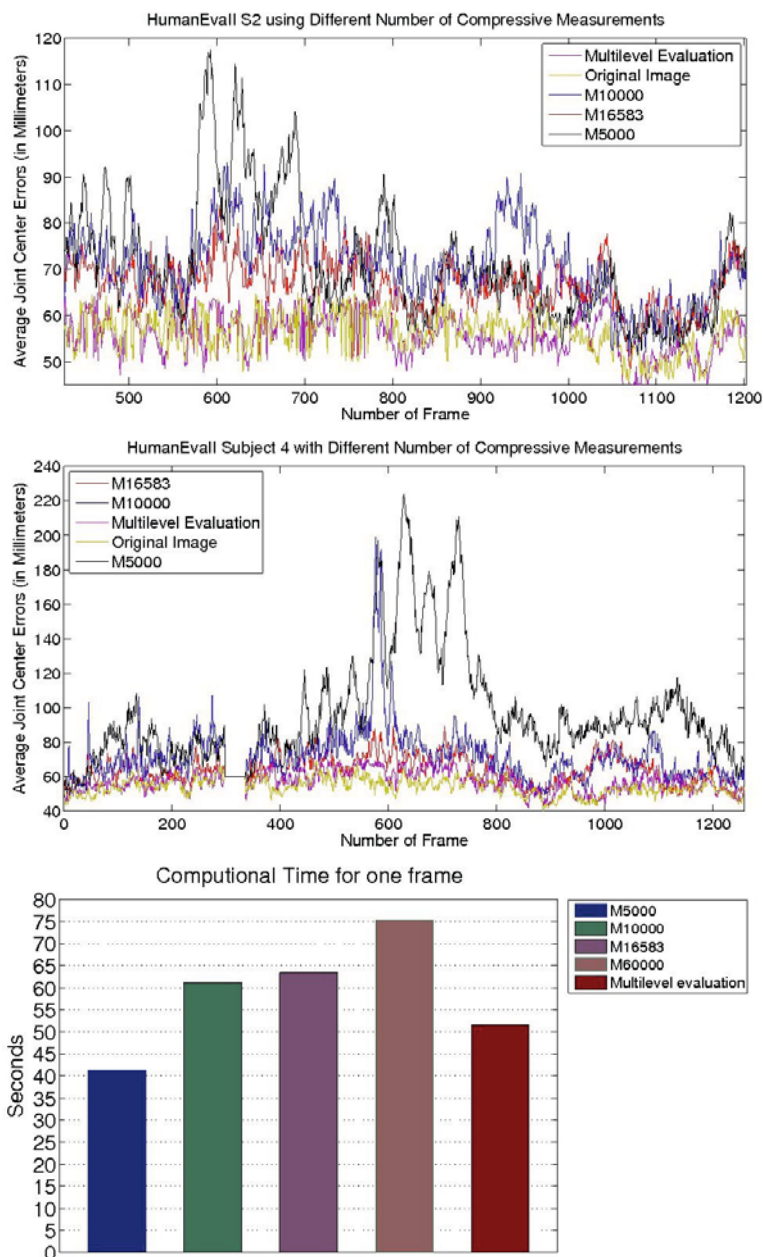


Fig. 4. From top to bottom, 1) tracking results of HumanEvaII Subject 2, 2) tracking results of HumanEvaII Subject 4 (the ground truth data is corrupted at 298-335 frames) and 3) computational time for one frame using the different number of compressive measurements



Fig. 5. HumanEvaII visual tracking results of Subject 4 and 2 are shown at the top four rows and the bottom four rows, respectively. The transparent visual model is overlapped with the tracking subject.

performance is degraded dramatically and we merely obtain $70.4249 \pm 7.5613mm$ and $68.2124 \pm 11.6153mm$, respectively. The middle of Figure 4 shows the experimental results of HumanEvaII Subject 4. The evaluation using original images achieves $54.2207 \pm 4.9250mm$ which is slightly better than $57.1705 \pm 6.0227mm$ achieved by the multilevel evaluation. Using $M = 16583$ compressive measurements experiences slightly more fluctuations comparing with the results of Subject 2. When the number of compressive measurements is decreased to $M = 10000$ and $M = 5000$, there are significant mistrackings and drifts with larger errors $71.6053 \pm 15.4005mm$ and $96.3663 \pm 32.8075mm$. More visual tracking results are shown in Figure 5.

The computational performance is also evaluated via the computational time for one frame using the different number of the compressive measurements shown in the bottom of Figure 4. As expected, the computational times from 40 to 75 seconds roughly correspond to increasing the number of the compressive

measurements M . On the other hand, the multilevel evaluation is able to reach the level of computational speed similar to merely using $M = 10000$ compressive measurements. Overall, the utilisation of progressive coarse-to-fine multilevel evaluation allows our approach to achieve the computational efficiency as only using $M = 10000$ compressive measurements and maintain the comparable tracking accuracy as using the original images.

6 Conclusion and Future Work

This paper has presented a compressive sensing framework for human tracking. It is realised by introducing a compressive observation model into the annealed particle filter. As the restricted isometry property ensures the preservation of the pairwise distance, compressive measurements with relative lower dimensions can be directly employed in observation evaluations without reconstructing the original image. Furthermore, noticing that there is a similar progressive process between the annealing schedule and the wavelet decomposition, we propose a novel multilevel wavelet likelihood evaluation in the coarse-to-fine fashion in which a fewer wavelet coefficients are used at the beginning, and then elevated gradually. This saves computational time and hence boosts the speed of evaluations. Finally, the robustness and efficiency of our approach are verified via the benchmark dataset HumanEvaII.

In compressive sensing recovery, many signal processing problems do not require full signal recovery and rather prefer to work on the compressive domain to benefit from dimensionality reduction. Indeed, RIP which approximately preserves an isometry allows to conduct evaluations and analysis on compressive measurements. However, the computational complexity of generating the sparse basis representation (in our case the wavelet decomposition) and compressive measuring still remains very high. In future work, we therefore would like to explore more about how to design more efficient the sparse basis representation and compressive measuring to handle the problem.

Acknowledgement. Authors would like to thank the support from National ICT Australia, and Leonid Sigal from Brown University provides the HumanEva dataset available.

References

1. Candes, E.J., Tao, T.: Decoding by linear programming. *IEEE Transactions on Information Theory* 51, 4203–4215 (2005)
2. Candès, E.J., Romberg, J.K., Tao, T.: Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory* 52, 489–509 (2006)
3. Candès, E.J., Romberg, J.K., Tao, T.: Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics* 59, 1207–1223 (2006)

4. Candes, E.J., Romberg, J.: Sparsity and incoherence in compressive sampling. *Inverse Problems* 23, 969–985 (2007)
5. Duarte, M.F., Davenport, M.A., Takhar, D., Laska, J.N., Sun, T., Kelly, K.F., Baraniuk, R.G.: Single-pixel imaging via compressive sampling. *IEEE Signal Processing Magazine* 25, 83–91 (2008)
6. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31, 210–227 (2009)
7. Cevher, V., Sankaranarayanan, A., Duarte, M., Reddy, D., Baraniuk, R., Chellappa, R.: Compressive sensing for background subtraction. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part II*. LNCS, vol. 5303, pp. 155–168. Springer, Heidelberg (2008)
8. Mei, X., Ling, H.: Robust visual tracking using l1 minimization. In: *ICCV 2009*, pp. 1436–1443 (2009)
9. Magnenat-Thalmann, N., Laperrière, R., Thalmann, D.: Joint-dependent local deformations for hand animation and object grasping. In: *Proceedings on Graphics Interface 1988*, Canadian Information Processing Society, pp. 26–33 (1988)
10. Natarajan, B.K.: Sparse approximate solutions to linear systems. *SIAM Journal on Computing* 24, 227–234 (1995)
11. Deutscher, J., Blake, A., Reid, I.: Articulated body motion capture by annealed particle filtering. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 126–133 (2000)
12. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. *Science* 220 (4598), 671–680 (1983)
13. Baron, D., Duarte, M.F., Wakin, M.B., Sarvotham, S., Baraniuk, R.G.: Distributed compressive sensing. The Computing Research Repository abs/0901.3403 (2009)
14. Baraniuk, R.G., Wakin, M.B.: Random projections of smooth manifolds. *Foundations of Computational Mathematics* 9, 51–77 (2009)
15. Johnson, W., Lindenstrauss, J.: Extensions of Lipschitz mappings into a Hilbert space. In: *Conference in modern analysis and probability (New Haven, Conn., 1982)*. *Contemporary Mathematics*, vol. 26, pp. 189–206. American Mathematical Society, Providence (1984)
16. Householder, A.S.: Unitary triangularization of a nonsymmetric matrix. *Journal of the ACM* 5, 339–342 (1958)
17. Sigal, L., Black, M.J.: Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. Technical report, Brown University, Department of Computer Science (2006)
18. Daubechies, I.: *Ten Lectures on Wavelets*. CBMS-NSF Regional Conference Series in Applied Mathematics. SIAM, Philadelphia (1992)

Reconstructing Mass-Conserved Water Surfaces Using Shape from Shading and Optical Flow

David Pickup, Chuan Li, Darren Cosker, Peter Hall, and Phil Willis

Media Technology Research Centre
University of Bath

{d.pickup,c1249,d.p.cosker,pmh,p.j.willis}@cs.bath.ac.uk

Abstract. This paper introduces a method for reconstructing water from real video footage. Using a single input video, the proposed method produces a more informative reconstruction from a wider range of possible scenes than the current state of the art. The key is the combination of vision algorithms and physics laws. Shape from shading is used to capture the change of the water's surface, from which a vertical velocity gradient field is calculated. Such a gradient field is used to constrain the tracking of horizontal velocities by minimizing an energy function as a weighted combination of mass-conservation and intensity-conservation. Hence the final reconstruction contains a dense velocity field that is incompressible in 3D. The proposed method is efficient and performs consistently well across water of different types.

1 Introduction

In recent years, fruitful progress has been made in reconstructing complex objects and scenes from images or videos, for example: faces [6], human bodies [1] hair [16], trees [21] [20] and fluids [2]. Among them, water brings unique challenges, a solution to which is of great interest to many research areas such as mechanical engineering [19] and computer graphics [23]. Traditional vision techniques are found to work less well in these cases. Major challenges include: a water surface generally lacks visually salient features; its complex dynamics, including topological changes, yield extreme difficulties for tracking; ground truth data is difficult to acquire – even active acquisition systems such as laser scanners will fail due to the over complicated reflection and refraction conditions.

This paper advances the current art of image based water reconstruction to work with a single input video captured in ordinary outdoor conditions, where the water is of a large scale and appears opaque. In these cases the traditional refraction and reflection based techniques as well as sophisticated experimental setups are impractical.

The proposed method is not only more flexible than previous methods of modelling the surface geometry but also reconstructs extra information in the form of a dense grid of 3D velocities. The key is the combination of shape from shading and optical flow using a physical constraint. First, shape from shading is used to estimate the geometry of the water surface for each frame. Although this

is an unusual method for reconstructing reflective and refractive materials, we will demonstrate in our experiments that the opaque appearance of large bodies of water outdoors, and our choice of shape from shading algorithm cause this method to produce a convincing result (figure 2). We then produce a vertical velocity gradient field calculated from the change of the recovered surface over time. This vertical gradient is coupled with the law of mass-conservation to constrain the tracking of horizontal velocities on the water surface. The final vertical velocity is recovered from the tracked horizontal velocities, producing the dense 3D velocity field.

Compared to the existing state of the art, the proposed method has the following advantages:

- It is designed to work with a single input video recorded by an ordinary capturing device. All the example videos are recorded by a digital video camera in an outdoor environment, where the water is of a large scale and appears opaque.
- It is more informative as not only the surface geometry is recovered, but so is a dense 3D velocity field.
- The recovered velocities comply with the conservation of mass in 3D.
- It is practically efficient and stable. No complex optimization schemes are used and experiments show it performs consistently well across different scenarios with fixed parameters.

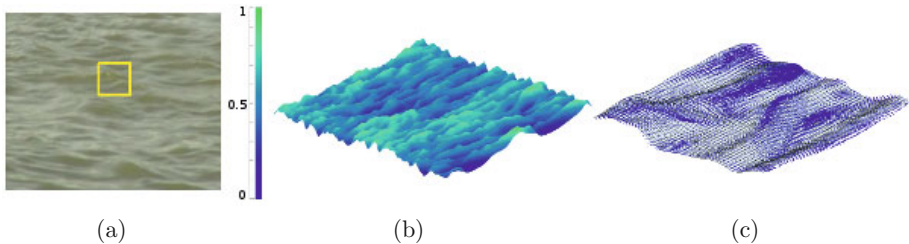


Fig. 1. The proposed single video based water reconstruction method. **a:** One frame from the input video. **b:** The fluid surface is recovered using combined shape from shading and optical flow. The surface geometry is demonstrated in 3D. **c:** Details of the 3D velocities and geometry inside the yellow box shown in (a). All height field results are normalised to $[0, 1]$ for visualization.

2 Related Work

The proposed method aims to reconstruct both the geometry and the velocity of the water. Two major research areas will be reviewed: water surface geometry reconstruction and fluid tracking.

2.1 Surface Geometry Reconstruction

Various types of physical properties have been used to reconstruct the water surface geometry, for example, refraction [3,12,13] and reflection [23], as well as others [10] [8].

Murase [13] reconstructs a water surface from the apparent motion of a refracted pattern. The distortion of an underwater pattern is tracked by optical flow, from which the water’s surface normal is calculated using a refraction model. The water surface is then recovered by 2D integration of the surface normal. Balschbach *et al.* [3] also use a refraction approach, but based on a shape from shading technique where multiple illuminations are used to better determine surface gradients. Morris and Kutulakos [12] show that refractive index is not indispensable by assuming light is refracted only once. Their system reconstructs the water surface by minimizing the refractive disparity. These refraction based methods are generally called “shape from distortion” and they work well for transparent water. The disadvantages are they can not work with opaque liquids and specially designed devices are required to capture the distortion of a known pattern being located underneath the surface of the water. These methods are not suitable for outdoor conditions where water often appears opaque.

Shape from stereo techniques have been explored to reconstruct liquids that are opaque. Wang *et al.* [23] dye water with white paint and light patterns are projected onto its surface. A depth field is first reconstructed by dense reconstruction and then refined using physically-based constraints. This method shows very accurate reconstructions of surface details. Ihrke *et al.* [10] dissolve the chemical Fluorescein in the water and measures the thickness of the water from the amplitude of the emitted light. The visual hull of the water surface is then calculated by utilizing weighted minimal surfaces using the thickness measurements as constraints. Hilsenstein [8] reconstructs water waves from thermographic image sequences acquired from a pair of infrared cameras. As a viable approach, infrared stereo reduces the problem associated with transparency, specular reflection and lack of texture at visible wavelengths. These techniques all require sophisticated equipment and complex experimental setups.

Missing from the literature is a solution for reconstructing water surfaces from a single video captured in an ordinary outdoor environment, as demonstrated by Figure 1(a). In this case, nothing can be put under or dissolved in the water. The water is almost opaque, where refraction based approaches are impracticable but reflection based approaches tend to gain performance. This paper demonstrates shape from shading is able to perform consistently well across different types of such water surfaces.

2.2 Fluid Tracking

Although surface geometry is important, it does not contain the full set of water properties. It only describes the change of the water surface height over time, while horizontal velocities are missing. Various types of trackers are proposed to acquire the fluid flow field.

Traditional 2D tracking algorithms such as Horn-Schunck optical flow [9] are found to perform less well for water where the conservation of intensity rarely holds. As an improvement, Nakajima *et al.* [14] propose an energy function as a weighted combination of conservation of intensity, conservation of mass, and momentum equations. The resulting flow complies with physical properties of fluids

in 2D. Doshi and Bors [5] use a robust kernel which adapts to the local data geometry in the diffusion stage of the Navier-Stokes formulation. The kernel ensures that smoothing occurs along the structure of the motion field while maintaining the general optical flow structure and the main optical flow features. Sakaino [18] proposes a method to model abrupt image flow change. Flow is modelled using a number of base waves and their coefficients are found to match the input sequence. Although these methods significantly improve 2D flow tracking, their physical constraints are not designed to work in 3D.

Papadakis *et al.* [15], and Heas and Memin [7] estimate 3D motions of a stratified atmosphere by minimizing an object function that describes the dynamics of an interacting stack of atmospheric layers. Li [11] treats the image as a wave-front surface and derives a general brightness constraint to model brightness variation in terms of fluid dynamics of the velocity potential. The gradient of the 3D velocity potential describes the actual motion flow. The general brightness constraint separates the flow dynamic from the brightness variation, hence one can replace the fluid dynamics model with other physical models and reuse the same solution process.

The method proposed in this paper recovers 3D velocities, producing a more informative reconstruction than previous 2D tracking algorithms. Compared with [7, 11, 15] the novelty of the proposed method is the combination of shape from shading and optical flow. Surface information acquired from the former is used as a prior to improve the performance of the latter, where physical rules are incorporated. The method is efficient and performs consistently well across different types of water captured in an outdoor environment.

3 Reconstructing 3D Mass-Conserved Water

The proposed method reconstructs the surface geometry and a dense 3D velocity field of water captured with a single video camera. The key is the law of mass-conservation, which is used as a physical link between the change of the surface height and the horizontal velocities. The proposed method uses shape from shading to acquire the change of surface height over time. It is then used as a prior to constrain the optical flow tracking. The final water surface will be reconstructed back from the horizontal velocities.

The rest of this section will first introduce the water model and the law of mass-conservation; then demonstrate shape from shading in acquiring surface geometry for a wide range of water; the physically constrained fluid tracking is explained at the end.

3.1 Conservation of Mass

A height field $h(x, y, t)$ is used to represent the water surface at time t . A vector $\mathbf{u} = (u, v, w)$ is used to represent the 3D velocity for each point on the surface. The law of mass-conservation constrains the 3D divergence of the velocity to zero, which leads to

$$\frac{\partial w}{\partial z} = -\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) \quad (1)$$

We first shown how the vertical velocity w can be approximated from the divergence of the horizontal velocities (u, v) . By assuming the horizontal velocities do not vary along the z -direction, the right-hand side of this equation does not depend on z , so $\frac{\partial w}{\partial z}$ is a constant along the z -direction. This means the vertical velocity w is a linear function of the water depth z . The velocity at the bottom of the water comes from the boundary condition $\mathbf{u} \cdot \mathbf{n} = 0$ where \mathbf{n} is the normal of the water bed. By further assuming a flat bottom, we have $\mathbf{n} = (0, 0, 1)$ hence w needs to be zero to satisfy the boundary conditions. Integrating $\frac{\partial w}{\partial z}$ along z -direction gives the vertical velocity:

$$w = h \frac{\partial w}{\partial z} = -h\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) \quad (2)$$

The vertical velocity can also be calculated from the material derivative of the surface height with respect to time:

$$w = \frac{dh}{dt} = \frac{\partial h}{\partial x}u + \frac{\partial h}{\partial y}v + \frac{\partial h}{\partial t} \quad (3)$$

Here we simplify the fluid dynamic by not considering the advection part $\frac{\partial h}{\partial x}u + \frac{\partial h}{\partial y}v$. Hence the Eulerian measurement of the surface change is used as an approximation of the vertical velocity $w \approx \frac{\partial h}{\partial t} = h(x, y, t + 1) - h(x, y, t)$. This significantly simplifies the later optimization process and experiments show the results are generally plausible.

The evolution of water surface can then be directly linked to horizontal velocities via:

$$h(x, y, t + 1) - h(x, y, t) = -h(x, y, t)\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) \quad (4)$$

Accurate horizontal velocities are expected to satisfy the surface change over time based on equation 4. The rest of this section first demonstrates shape from shading can be used to acquire a prior for water surface and then explains how to use such a prior to improve the tracking of horizontal velocities.

3.2 Recovering the Water Surface Using Shape from Shading

Shape from shading deals with the recovery of shape from a gradual variation of shading in the image, see Zhang *et al.* [24] for a detailed survey. A general assumption made by shape from shading techniques is that the scene follows the Lambertian model, in which the grey level at a pixel in the image depends on the light source direction and the surface normal. For specular surfaces, this assumption holds less well and more complex reflection/refraction models [4] are expected to be needed.

Although water is expected to be a highly reflective and refractive substance, we show that shape from shading can provide a high quality reconstruction of an outdoor water surface. Figure 2 shows eight scenes captured in ordinary outdoor conditions with their shape from shading recovered surfaces underneath (using Tsai *et al.*'s method [22]). One important reason for shape from shading to perform so well is that the water in these scenes appears visually opaque because of its depth and the suspension of dirt, mud and air. Also the particular shape from shading algorithm [22] used here is reported to have good performance with specular surfaces.

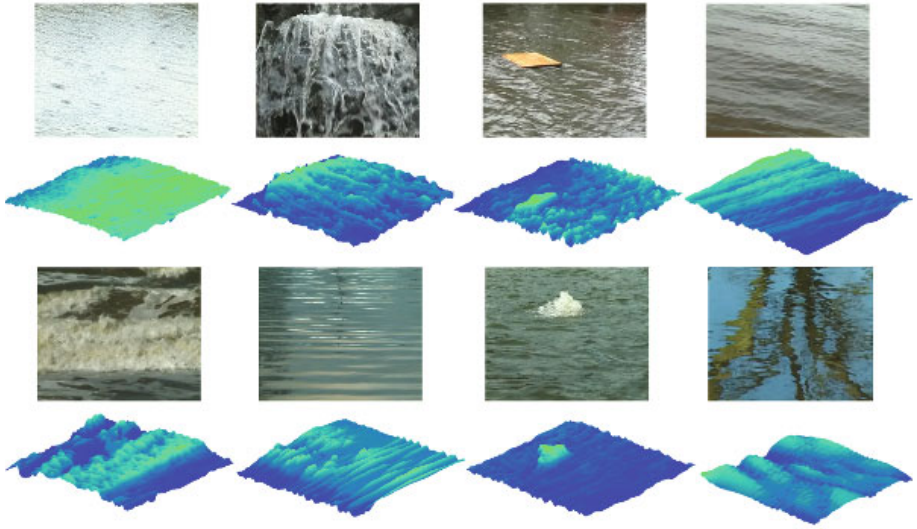


Fig. 2. Despite distortions from strong reflections (bottom right), experiments show shape from shading performs consistently well in recovering water surfaces of different types

Our experiments also show shape from shading can work for dynamic water with very few adaptations. Videos are low-pass filtered to remove noise, such as extreme bright or dark points. A height field $h(x, y, t)$ is then individually recovered for each frame t to represent the water surface. For a T -frames video of resolution M by N , the average height of each surface $\frac{1}{MN} \sum_{i=1}^N \sum_{j=1}^M h(i, j, t)$ is rectified to the same level $\frac{1}{TMN} \sum_{k=1}^T \sum_{i=1}^N \sum_{j=1}^M h(i, j, k)$ to remove the affect of global luminance change:

$$h'(x, y, t) = h(x, y, t) - \frac{1}{MN} \sum_{i=1}^N \sum_{j=1}^M h(i, j, t) + \frac{1}{TMN} \sum_{k=1}^T \sum_{i=1}^N \sum_{j=1}^M h(i, j, k) \quad (5)$$

An example is shown in figure 3, where the shape from shading surface successfully follows the movement of the water in the video. However, surface geometry

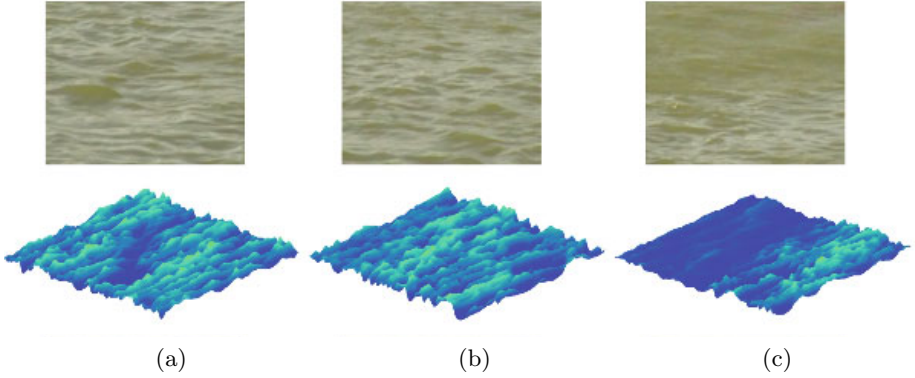


Fig. 3. Experiments show shape from shading is able to reconstruct the change of the fluid surface over time. **a - c:** different frames in the sequence and their shape from shading reconstructions.

is not a completely informative description for water, but can be used to constrain optical flow to obtain the velocities of the surface using the law of mass conservation.

3.3 Combined Shape from Shading and Optical Flow

The general idea is to use shape from shading water surfaces to constrain the tracking of horizontal velocities based on the conservation of mass. As explained in section 3.1, the vertical velocity w is approximated as the Eulerian derivatives of the shape from shading surfaces with respect to time. Its gradient along the z -direction $\frac{\partial w}{\partial z}$ is consequently calculated as $\frac{h(x,y,t+1)-h(x,y,t)}{h(x,y,t)}$. The horizontal velocities (u, v) are then whatever it takes to make the water incompressible.

The objective energy function is a weighted combination of intensity-conservation, mass-conservation and smoothness:

$$E = \int \int [(I_x u + I_y v + I_t)^2 + \alpha^2 (|\nabla u|^2 + |\nabla v|^2) + \beta^2 (u_x + v_y + w_z)^2] dx dy \quad (6)$$

$(I_x u + I_y v + I_t)^2$ and $|\nabla u|^2 + |\nabla v|^2$ are the intensity-conservation term and smoothness terms from the Horn-Schunck [9] optical flow. $(u_x + v_y + w_z)^2$ is the mass-conservation term that describes the 3D divergence of the velocity. In practice, w is calculated by subtracting the current shape from shading surface from its successor. Then w_z is calculated as $\frac{w}{h}$. The following Euler-Lagrangian equations are used to minimize the objective function [6]:

$$I_x(I_x u + I_y v + I_t) - \alpha^2 \Delta u - \beta^2 (u_{xx} + v_{xy} + w_{xz}) = 0 \quad (7)$$

$$I_y(I_x u + I_y v + I_t) - \alpha^2 \Delta v - \beta^2 (u_{xy} + v_{yy} + w_{yz}) = 0 \quad (8)$$

In practice Δu , Δv , u_{xx} and v_{yy} are approximated numerically using finite differences: $\tilde{u}(x, y) - u(x, y) = \frac{u(x-1, y) + u(x+1, y) + u(x, y-1) + u(x, y+1) - 4u(x, y)}{4}$, $\tilde{v}(x, y) - v(x, y) = \frac{v(x-1, y) + v(x+1, y) + v(x, y-1) + v(x, y+1) - 4v(x, y)}{4}$, $\bar{u}(x, y) - u(x, y) = \frac{u(x-1, y) + u(x+1, y)}{2} - u(x, y)$, $\bar{v}(x, y) - v(x, y) = \frac{v(x, y-1) + v(x, y+1)}{2} - v(x, y)$. The Lagrange multipliers α^2 and β^2 are fixed to 1000 across all scenes. The solution of equations 7 and 8 is found using the Gauss Seidel method. The resulting horizontal velocity (u, v) is then used to calculate the final vertical velocity w and the change of the water surface using equation 4. Due to the mass conservation constraint the surface produced from these new vertical velocities is very similar to the shape from shading surfaces, which have been shown to model the real water dynamics well.

4 Experiment

To evaluate the quality of our method we compare our method with several state of the art flow estimators on different water scenes. Our hypothesis is that our method will track the horizontal flow of the fluid more plausibly than previous methods, the major improvement being that our result conforms with the movement of fluid in 3D. We compare both the appearance of the tracked horizontal velocities alone, and the surface reconstructed using mass-conservation.

This paper chooses the classical Horn-Schunck [9] optical flow and the more contemporary physics-based flow tracker [14] to compare with. These two methods, like ours, both minimize an energy function as the weighted combination of some energy terms such as the intensity-conservation term and the smoothness term. The difference is Nakajima *et al.*'s [14] method contains extra terms for 2D momentum equations and 2D mass-conservation; the proposed method in this paper contains an extra term for mass-conservation in 3D and Horn-Schunck [9] flow does not employ any physical constraint. In this paper, same weight coefficients (Lagrangian multipliers) are used to combine different energy terms and they are fixed across all the experiment sequences.

Figure 4 shows the horizontal flow fields acquired using the three different methods. The flow produced by the Horn-Schunck [9] method clearly oversmooths the velocities and only captures the global flow of the different image regions. The flow field produced by Nakajima *et al.* [14] improves on this but still oversmooths the finer details of the water movement. As demonstrated, our method manages to create a flow field which captures the detailed sharp features of the flow successfully.

We have produced reconstructions of the surface geometry using the velocities by both the Horn-Schunck [9] and Nakajima *et al.* [14] methods. Figure 5 shows the surface geometries produced using these vertical velocities and an initial height at frame 1 produced by shape from shading. This experiment evaluates how well the velocities produced by each algorithm comply with the movement of fluid in 3D. Our results show that both methods tend to “halt” the water surface due to the lack of vertical velocity. As error accumulates in time, the water surface drifts away from its real appearance in the video. These results

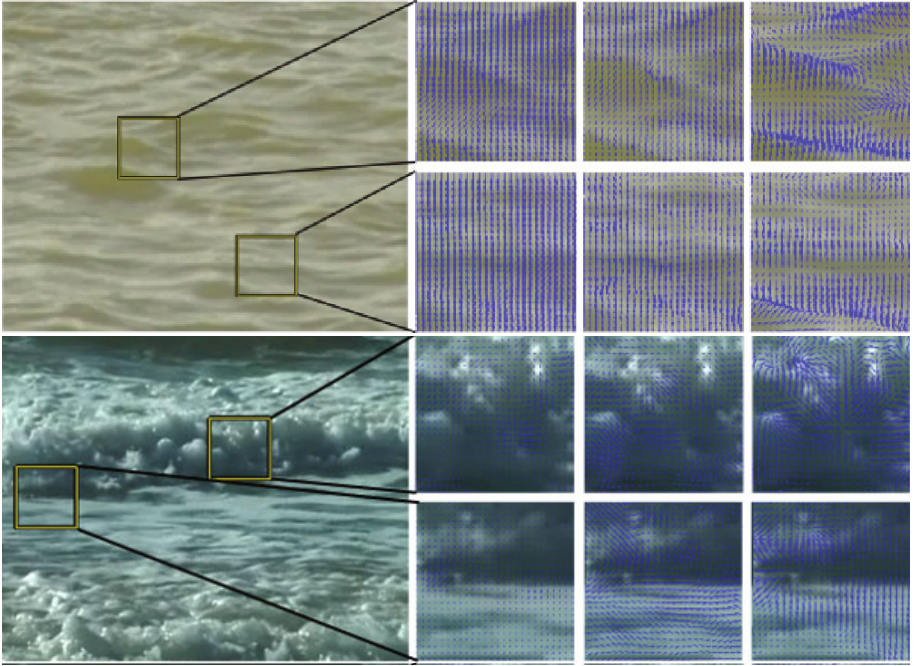
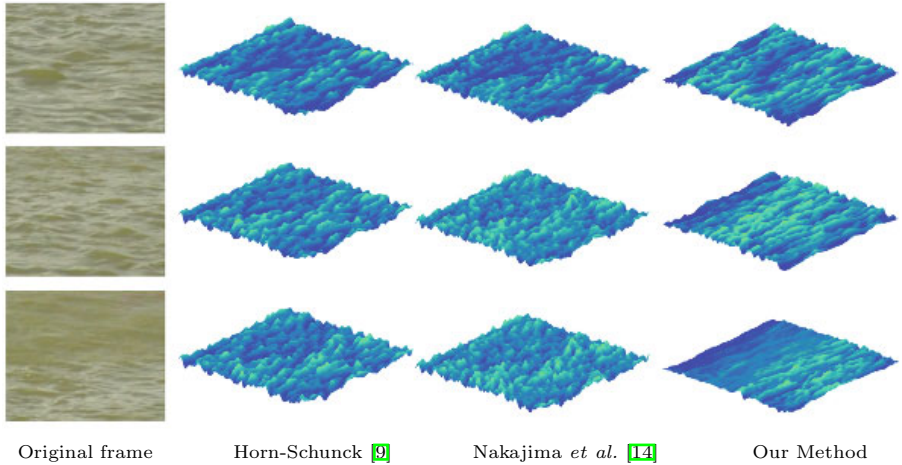
Horn-Schunck [9] Nakajima *et al.* [14] Our Method

Fig. 4. Results of different methods. Our method successfully captures the sharp velocity features, while previous methods tend to over smooth the flow.



Original frame

Horn-Schunck [9]

Nakajima *et al.* [14]

Our Method

Fig. 5. Results of reconstructions produced from horizontal velocities given by different flow estimators. The Horn-Schunck and Nakajima reconstructions are “halted” and noisy, while the proposed method is significantly better.

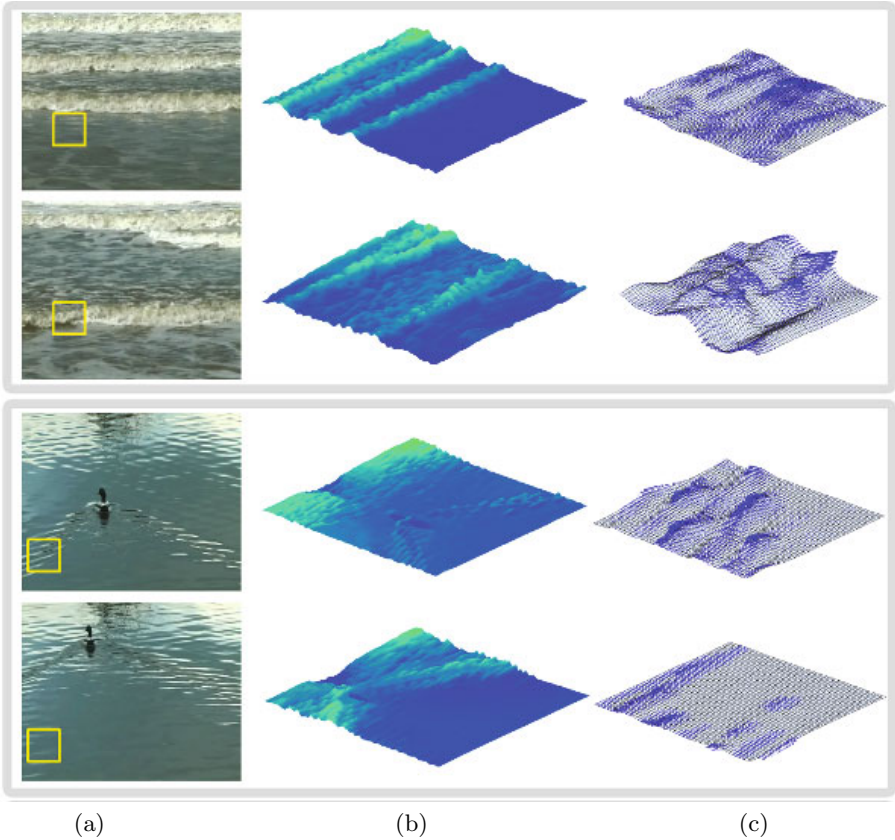


Fig. 6. Results of different water surfaces. **a:** the original input video frame. **b:** the mass-conserved surface reconstructions. **c:** 3D velocities and geometry of the surface inside the yellow box shown in (a). Each pair of results are two frames from the same video sequence.

are due to the lack of a 3D physical constraint and therefore the vertical velocities calculated using the 3D law of mass-conservation are incorrect.

The robustness of the proposed method is tested on a wide range of water sequences. 40 water sequences from the Dyntex database [17] are used. These include water of calm, wavy and turbulent motion. The sequences are filmed outdoors with an ordinary digital video camera with a fixed tripod. A common property of these videos is the water generally appears opaque which allows the shape from shading surface a veridical prior to constrain the optical flow tracker. Results show the proposed method performs consistently across these test sequences. Some of the reconstructed water surfaces and velocity fields are shown in figure 6. The fluid dynamics caused by objects interfering, such as an animal swimming, can also be well captured.

An advantage of the proposed method is its efficiency. Solving equations 7 and 8 is a linear optimization process without any extra complexity compared

to the classic Horn-Schunck [9] optical flow. A C++ implementation of the whole system, including shape from shading and flow estimation, is able to process over 10 frames of resolution 352×288 per second on an Intel quad-core processor, which makes realtime applications practically possible.

There are several limitations of the proposed method. First, it strongly depends on the surface prior acquired from shape from shading. Although it has been shown in this paper that shape from shading works consistently well over a wide range of water that has opaque and Lambertian properties, failure modes can appear when the water is transparent or highly specular. In this case the refraction/reflection will distort the reconstructed surface. A good example is shown in the last picture of figure 2 where the reflection of the trees yield valleys on the surface. Currently the proposed method simply uses a low-pass filter to remove the extreme bright or dark pixels in the image, this can be replaced by better specular/shadow removal methods. Also, the height field representation works efficiently well for calm water surfaces but does not well describe complex scenes such as splashing and breaking waves. In these cases a more sophisticated fluid representation is needed to handle the topological change.

In summary an important characteristic of the reconstruction is it is physically sound, as the velocity field complies with the conservation of mass in 3D. Compared to previous flow estimators our method captures sharp velocity features and reconstructs a water surface that successfully models the change of the water surface geometry. Our method works fully automatically and requires only a single input video. It has been tested on a wide range of scenes and found to perform consistently (figure 6).

5 Conclusion

This paper studied the problem of image-based water reconstruction for a single input video that is captured in ordinary outdoor conditions. In this case the water is of a large scale, appears opaque and traditional refraction and reflection based reconstruction techniques are impractical. One important discovery is the capability of shape from shading to recover different water surfaces of this kind. Consistent performance is demonstrated by experimenting on a wide range of scenes. Based on this discovery, the paper proposes a method for reconstructing water by combining shape from shading and optical flow. It essentially uses the vertical velocity acquired from shape from shading to constrain the optical flow tracking of horizontal velocities. The advantages of the proposed method are: 1) it works fully automatically and requires only basic input resources; 2) the reconstruction is more informative as it contains not only the surface geometry profile but also a 3D velocity field; 3) the recovered velocities are mass-conserved in 3D; 4) it is efficient and generally stable, as tested by a wide range of water. We also discussed several failure modes where the water is highly specular. Interesting future avenues include finding better solutions for removing shadows and highlights from the water surface and integrating more sophisticated fluid dynamics, for example the full Navier-Stokes equations.

Acknowledgments. We wish to thank Yi-Zhe Song, Liang Wang, Tom Saunders and Marios Richards for their comments and suggestions. We would also like to thank the University of Bath and the Centre for Digital Entertainment for funding this project.

References

1. Aguiar, E.D., Stoll, C., Theobalt, C., Ahmed, N., Seidel, H., Thrun, S.: Performance capture from sparse multi-view video. In: Proceedings of ACM SIGGRAPH, vol. 27, pp. 1–10 (2008)
2. Atcheson, B., Ihrke, I., Heidrich, W., Tevs, A., Bradley, D., Magnor, M., Seidel, H.: Time-resolved 3d capture of non-stationary gas flows. In: Proceedings of ACM SIGGRAPH Asia, vol. 27, pp. 1–9 (2008)
3. Balschbach, G., Klinke, J., Jähne, B.: Multichannel shape from shading techniques for moving specular surfaces. In: Burkhardt, H., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1407, pp. 170–184. Springer, Heidelberg (1998)
4. Ding, Y.Y., Yu, J.Y., Sturm, P.: Recovering specular surfaces using curved line images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2326–2333 (2009)
5. Doshi, A., Bors, A.G.: Navier-stokes formulation for modelling turbulent optical flow. In: Proceedings of the British Machine Vision Conference, pp. 1–10 (2007)
6. Ghosh, A., Hawkins, T., Peers, P., Frederiksen, S., Debevec, P.: Practical modeling and acquisition of layered facial reflectance. In: Proceedings of ACM SIGGRAPH Asia, vol. 27, pp. 1–10 (2008)
7. Héas, P., Mémin, E.: Three-dimensional motion estimation of atmospheric layers from image sequences, vol. 46, pp. 2385–2396 (2008)
8. Hilsenstein, V.: Surface reconstruction of water waves using thermographic stereo imaging. In: Image and Vision Computing, New Zealand, pp. 102–107 (2005)
9. Horn, B.K.P., Schunck, B.G.: Determining optical flow. *Artificial Intelligence* 17, 185–203 (1981)
10. Ihrke, I., Goldluecke, B., Magnor, M.: Reconstructing the geometry of flowing water. In: Proceedings of the International Conference on Computer Vision, pp. 1055–1060 (2005)
11. Li, F., Xu, L.W., Guyenne, P., Yu, J.Y.: Recovering fluid-type motions using navier-stokes potential flow. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2010)
12. Morris, N.J., Kutulakos, K.N.: Dynamic refraction stereo. In: Proceedings of the International Conference on Computer Vision, pp. 1573–1580 (2005)
13. Murase, H.: Surface shape reconstruction of a nonrigid transport object using refraction and motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14, 1045–1052 (1992)
14. Nakajima, Y., Inomata, H., Nogawa, H., Sato, Y., Tamura, S., Okazaki, K., Torii, S.: Physics-based flow estimation of fluids. *Pattern Recognition* 36, 1203–1212 (2003)
15. Papadakis, N., Héas, P., Mémin, E.: Image assimilation for motion estimation of atmospheric layers with shallow-water model. In: Proceedings of the Asia Conference on Computer Vision, pp. 864–874 (2007)
16. Paris, S., Chang, W., Kozhushnyan, O.I., Jarosz, W., Matusik, W., Zwicker, M., Durand, F.: Hair photobooth: geometric and photometric acquisition of real hairstyles. In: Proceedings of ACM SIGGRAPH, pp. 1–9. ACM, New York (2008)

17. Péteri, R., Fazekas, S., Huiskes, M.J.: Dyntex: a comprehensive database of dynamic textures. *Pattern Recognition Letters* (2010)
18. Sakaino, H.: Motion estimation method based on physical properties of waves. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1–8 (2008)
19. Shand, T., Shand, R., Bailey, D., Andrews, C.: Wave deformation in the vicinity of a long ocean outfall at wanganui, new zealand. In: *Coasts and Ports Australasian Conference*, pp. 173–178 (2005)
20. Tan, P., Fang, T., Xiao, J.X., Zhao, P., Quan, L.: Single image tree modeling. In: *Proceedings of ACM SIGGRAPH, Asia*, vol. 27, pp. 1–7 (2008)
21. Tan, P., Zeng, G., Wang, J.D., Kang, S.B., Quan, L.: Image-based tree modeling. In: *Proceedings of ACM SIGGRAPH*, vol. 87. ACM, New York (2007)
22. Tsai, P., Shah, M.: Shape from shading using linear approximation. *Image and Vision Computing* 12, 487–498 (1994)
23. Wang, H.M., Liao, M., Zhang, Q., Yang, R.G., Turk, G.: Physically guided liquid surface modeling from videos. In: *Proceedings of ACM SIGGRAPH*, pp. 1–11 (2009)
24. Zhang, R., Tsai, P.S., Cryer, J.E., Shah, M.: Shape from shading: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21, 690–706 (1999)

Earth Mover's Morphing: Topology-Free Shape Morphing Using Cluster-Based EMD Flows

Yasushi Makihara and Yasushi Yagi

Osaka University, 8-1 Mihogaoka, Ibaraki, Osaka, 567-0047, Japan

Abstract. This paper describes a method for topology-free shape morphing based on region cluster-based Earth Mover's Distance (EMD) flows, since existing methods for closed curve/surface-based shape morphing are inapplicable to regions with different genera. First, the shape region is decomposed into a number of small clusters by Fuzzy C-Means clustering. Next, the EMD between the clusters of two key shapes is calculated and the resultant EMD flows are exploited as a weighted many-to-many correspondence among the clusters. Then, the fuzzy clusters are transported based on the EMD flows and a transition control parameter. Unlike the closed curve/surface-based methods, the morphs using cluster transportation are not guaranteed to be a binary image, and hence graph cut-based binary denoising is applied to a volumetric image of the two-dimensional position and the one-dimensional transition control parameter. The experiments demonstrate that the proposed method can perform morphing between shapes with different genera, such as walking silhouettes or alphabetical characters.

1 Introduction

For a long time, image morphing [1] has attracted much attention in the image processing and computer graphics fields, because it serves as a powerful image/video editing tool for creating unique visual effects in view morphing [2] and 3D face synthesis [3]. Image morphing techniques are further used in computer vision and pattern recognition areas to generate view-interpolated images for efficient supervised learning [4] and training samples for deformable shape matching [5] [6].

In the early stages of morphing research, correspondences between geometric primitives including points, lines, and curves were manually given and various types of warping functions were proposed, such as mesh warping [7], field morphing [8], the radial basis function [9], thin plate spline [10], energy minimization-based function [11], and Multilevel Free-Form Deformation (MFFD) [12] [13]. During the middle stages, methods for automatic correspondence were proposed to reduce burdens of user input [14] [15] [16] [17] [18]. As the above methods rely on image texture, they are not applicable to shape morphing without texture.

On the other hand, the shape morphing problem is treated predominantly as a shape contour/surface deformation problem because of its compact expression [19] [20]. The morphing target is, however, limited to shapes with the same *genus*; in other words, most of the existing methods cannot deal with morphing

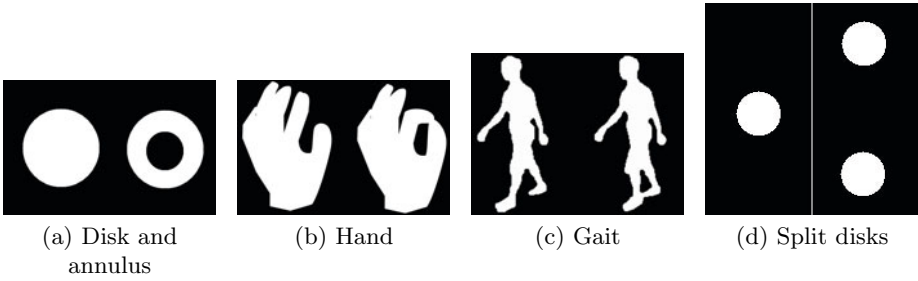


Fig. 1. Examples of pairs of shapes: (a), (b) and (c) with different genera (left: 0 genera, right: 1 genus), and (d) with a different number of shapes

of shapes with different genera, such as morphing from a disk to an annulus in a 2D domain (Fig. 1(a)) or from a ball to a torus in a 3D domain. Indeed, this limitation is critical for many silhouette-based applications including hand posture recognition (Fig. 1(b)), gait recognition (Fig. 1(c)), and action recognition. Although several methods [21] [22] can treat such topology differences, they suffer still from a tedious process which requires significant user input.

To cope with such topological changes, volume-based approaches were proposed and they generally fall into two categories, distance field approaches [23] [24] and level-set approaches [25] [26].

The distance field approaches first constructs a signed distance field to contour/surface for each shape and then generates an intermediate contour/surface by interpolating the signed distance fields. It is, however, reported that the distance field approaches sometimes produce undesirable pop-up artifacts [25]. For example, in case of morphing from a single disk to two distantly split disks (Fig. 1(d)), while the center disk disappears by erosion, the two split disks emerge as points and are dilated to the destination disks.

The level-set approaches also constructs the signed distance field and then a contour/surface is evolved based on a partial differential equation within so-called "narrow band" in the Level-Set Method (LSM) [27]. Because the narrow band gradually moves in the evolution process and it never pops up from the other region, it sometimes fails in reaching the destination shape. For example, in case of morphing from a single disk to two distantly split disks (Fig. 1(d)), the two split disks never emerge because the narrow band disappears after the center disk erosion and does not pop up from a region of the two split disk.

Consequently, we propose a method for topology-free shape morphing based on region cluster-based EMD flows in a 2D domain. The shape region is first decomposed into a set of small clusters and then EMD flows between the clusters of two key shapes are calculated. Each cluster is morphed and blended according to the many-to-many correspondence of the EMD flows and transition control parameter. Since the proposed method relies on the EMD flows, that is, a kind of warping motion, it is applicable not only to shapes with different topologies, but also to a different number of shapes as in (Fig. 1(d)), which results in more interesting split-process morphing as shown in Fig. 10.

2 Related Work

Automatic morphing of textured image: Gao et al. [14] proposed an energy minimization approach based on image feature consistency and deformation amount, while Tekalp [16] and Toshev et al. [15] exploited optical flow-based feature correspondences and saliency region correspondences, respectively. Chen et al. [17] exploited pixel correspondence based on range data and camera pose for view interpolation. Shechtman et al. [18] proposed a regenerative morphing from small pieces of the two source images based on source similarity and time coherence. Zhu et al. [28] [29] formulated morphing of textured images as optimal mass transportation problem and solved it in iterative energy minimization framework.

Polygonal/polyhedral shape morphing: Sederberg [19] proposed a 2D polygonal shape morphing method based on work minimization of the vertex deformation. Kent et al. [20] extended the idea to 3D polyhedral shape and computed a transformation process by interpolating between corresponding vertex positions. These methods is, however, not applicable to free-form shape.

Non-rigid shape matching and registration: In addition to pure morphing techniques, it is possible to include shape contour matching techniques, such as those based on geodesic distance [30] or the Earth Mover’s Distance (EMD) [31], in the shape morphing, as the matching results give the correspondence of each point on the contour. Non-rigid shape registration is also related to contour/surface-based shape morphing. Non-rigid shape is usually expressed as line segments for a 2D shape or surface meshes for a 3D shape and the correspondences of the contour/surface between two shapes are obtained in the registration process based on minimum distortion criteria [32], a data-driven deformation prior [33], or a elastic convolved ICP [34]. Then, interpolated non-rigid shapes can be generated based on the correspondences. These methods are, however, not applicable to shapes with different genera.

3 Earth Mover’s Morphing

3.1 Construction of Floating-Bin Histogram

The first step involves constructing a floating-bin histogram from a shape silhouette. First, the 2D position in the image is defined as $\mathbf{x} = [x, y]^T$ and subsequently, a silhouette image $I(\mathbf{x})$ is defined as

$$I(\mathbf{x}) = \begin{cases} 1 & \text{for inside shape} \\ 0 & \text{for outside shape.} \end{cases} \quad (1)$$

The shape silhouette can also be expressed as a set of points within the shape as $X_s = \{\mathbf{x} | I(\mathbf{x}) = 1\}$. In addition, an area $A(I) = \sum_{\mathbf{x}} I(\mathbf{x})$ and area-normalized weight $w(\mathbf{x}) = I(\mathbf{x})/A(I)$ are calculated in preparation of our formulation.

Next, N_c fuzzy clusters are obtained by Fuzzy C-Means (FCM) clustering [35]. The reason that fuzzy clusters are chosen instead of a hard clustering method

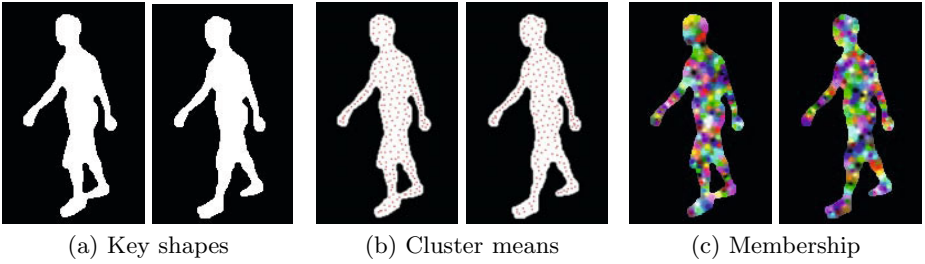


Fig. 2. Results of Fuzzy C-Means clustering. In each pair, the left and right images are the source and destination, respectively. Cluster mean positions are depicted as a red cross in (b). In (c), each color corresponds to a cluster and the membership for all the clusters is depicted with alpha blending.

such as k-means clustering, is the effectiveness of the fuzzy property in the denoising process in the final step described in [3.4](#).

Let the c th cluster's mean, weight, and membership at \mathbf{x} be $\bar{\mathbf{x}}_c$, \bar{w}_c , and $m_c(\mathbf{x})$, respectively, which satisfy

$$\bar{\mathbf{x}}_c = \frac{\sum_{\mathbf{x} \in X_s} m_c(\mathbf{x}) \mathbf{x}}{\sum_{\mathbf{x} \in X_s} m_c(\mathbf{x})} \quad (2)$$

$$w_c = \frac{\sum_{\mathbf{x} \in X_s} m_c(\mathbf{x}) w(\mathbf{x})}{\sum_{\mathbf{x} \in X_s} m_c(\mathbf{x})} \quad (3)$$

$$\sum_{c=1}^{N_c} m_c(\mathbf{x}) = 1. \quad (4)$$

Thus, a floating-bin histogram is composed of a set of bin means $\bar{\mathbf{X}} = \{\bar{\mathbf{x}}_c\}$ and a set of bin weights $\mathbf{w} = \{w_c\}$. Examples of the FCM clustering results are shown in [Fig. 2](#).

3.2 Acquisition of EMD Flow

The second step involves acquiring EMD flows from a source to a destination shape. Let sets of histogram bin means and weights for the source shape be $\bar{\mathbf{X}}^s, \mathbf{w}^s$ and those for the destination be $\bar{\mathbf{X}}^d, \mathbf{w}^d$, respectively. Then, the transportation cost and a flow (transportation amount) from the j th bin of the source shape to the k th bin of the destination shape are denoted as t_{jk} and f_{jk} , respectively.

Though the transportation cost is typically defined as the Euclidean distance between the bin means $d_{jk} = \|\mathbf{x}_j^s - \mathbf{x}_k^d\|$, this sometimes induces an inhomogeneous work assignment, that is, a situation in which the transportation distances of a few flows are too long, while the majority of the others are relatively short. As this inhomogeneity is undesirable in morphing in particular, we use the squared Euclidean distance $t_{jk} = d_{jk}^2$ instead. Since the squared distance is more sensitive to a distant transportation, the transportation distances tend to be similar to one another, in other words, the clusters tend to move closer together.



Fig. 3. NOR region-crossing distance

Moreover, when a large deformation is necessary in the morphing process, it often happens that several flows cross a *NOR* (Not-OR) region as shown in Fig. 3. Although it depends on the particular situation whether or not crossing the NOR region is undesirable, this can be suppressed by adding a NOR region-crossing distance to the transportation cost as

$$t_{jk} = d_{jk}^2 + \lambda^{NOR} d_{jk}^{NOR^2}, \tag{5}$$

where λ^{NOR} is a coefficient of the NOR region-crossing distance.

Finally, the EMD flows are optimized in the following framework in conjunction with the Hungarian algorithm.

$$\begin{aligned} \{f_{jk}\}^* &= \arg \min_{\{f_{jk}\}} \sum_j \sum_k f_{jk} t_{jk} & (6) \\ \text{s.t.} \quad & \sum_k f_{kl} = w_k^s \quad \forall k \\ & \sum_l f_{kl} = w_l^d \quad \forall l \\ & f_{kl} \geq 0 \quad \forall k, l \end{aligned}$$

Now, we can regard the obtained $\{f_{jk}\}^*$ as the cluster-based many-to-many warping weight coefficients, whereas most of the existing methods use one-to-one warping functions. Examples of the EMD flows and mean flows obtained from each cluster are shown in Fig. 4(a)(b).

3.3 Cluster-Based Morphing

The third step is the cluster-based morphing process using the obtained EMD flows $\{f_{jk}\}^*$. First, we consider a morphing from the j th cluster in the source shape to the k th cluster in the destination shape at a transition rate α as shown in Fig. 5. Next, let the interpolated position at the transition rate α between the j th cluster mean \bar{x}_j^s and the k th cluster mean \bar{x}_k^d be

$$\bar{x}_{jk}(\alpha) = (1 - \alpha)\bar{x}_j^s + \alpha\bar{x}_k^d \tag{7}$$

Then, suppose that a rate (f_{jk}/w_j^s) of the j th cluster is planned to be transported forward from x_j^s to x_k^d and that it is *dropped at the interpolated point* $\bar{x}_{jk}(\alpha)$ on

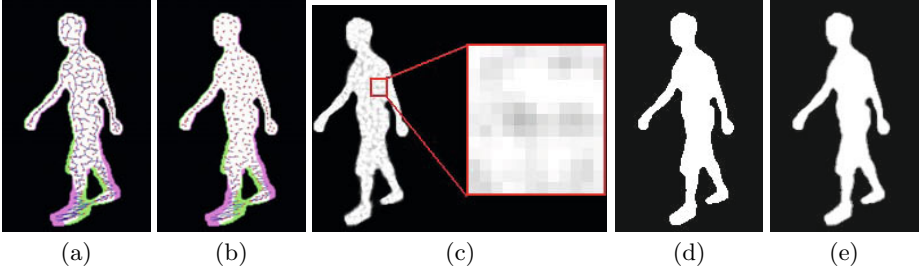


Fig. 4. EMD flows and morphing process between the key shapes in Fig. 2(a). (a) Raw EMD flows. (b) Mean flow of each cluster is calculated as a mean motion vector weighted by flow amount for visibility. (c) Blended morphing with artifacts. (d) Denoised binary morphing. (e) Boundary-dithered morphing.

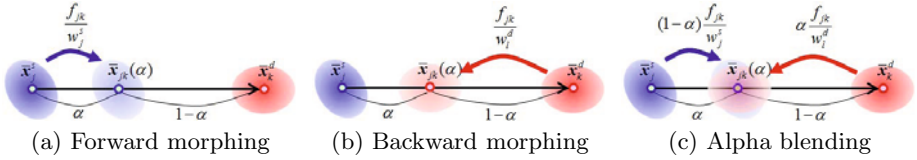


Fig. 5. Cluster-based morphing

the way in the forward transportation as shown in Fig. 5(a). In this paper, we call this process *forward morphs*. The forward morphs for all the EMD flows are blended to create a forward morphing image $I_{morph}^F(\mathbf{x}; \alpha)$. In the same way, a backward morphing image $I_{morph}^B(\mathbf{x}; \alpha)$ is created (Fig. 5(b)) and then the forward morphing image $I_{morph}^F(\mathbf{x}; \alpha)$ and backward morphing image $I_{morph}^B(\mathbf{x}; \alpha)$ are alpha-blended to create a blended morphing image (Fig. 5(c)) as

$$I_{blend}(\mathbf{x}; \alpha) = (1 - \alpha)I_{morph}^F(\mathbf{x}; \alpha) + \alpha I_{morph}^B(\mathbf{x}; \alpha) \quad (8)$$

3.4 Graph-Cut Denoising

The last step is the denoising process for the blended morphing image $I_{blend}(\mathbf{x}; \alpha)$. Unfortunately, the blended morphing image obtained by the cluster-based method suffers from "artifacts", that is, non-uniform silhouette intensity as shown in Fig. 4(c). Therefore, graph-cut denoising is applied to a volumetric blended morph image $I_{blend}(\mathbf{x}; \alpha)$ with 2D spatial positions and a 1D transition parameter α to create a volumetric binary image $I_{bin}(\mathbf{x}; \alpha)$. Let a three-dimensional site and its label be $\mathbf{u} = [\mathbf{x}^T, \alpha]^T$ and $l_{\mathbf{u}}$, respectively. In this paper, the label is set to 1 for the silhouette region and to 0 otherwise.

Now, graph-cut denoising is formulated as the following energy minimization problem.

$$E(L) = \sum_{\mathbf{u} \in U} g_{\mathbf{u}}(l_{\mathbf{u}}) + \sum_{(\mathbf{u}, \mathbf{v}) \in V} h_{\mathbf{u}\mathbf{v}}(l_{\mathbf{u}}, l_{\mathbf{v}}), \quad (9)$$

where L is a combination of labels for each site, U is the set of all sites, and V is all the combinations of neighborhood sites. The first term is referred to as the data term and the second term as the smoothness term. The data term is determined based on the pixel intensity of the blended morphing image as

$$g_v(l) = \begin{cases} 1 - I_{blend}(\mathbf{x}; \alpha) & (l = 0) \\ I_{blend}(\mathbf{x}; \alpha) & (l = 1) \end{cases} \quad (10)$$

The smoothness term is formulated by the Potts model as

$$h_{uv}(l_u, l_v) = \lambda_{potts}(1 - \delta_{l_u l_v}), \quad (11)$$

where λ_{potts} is the smoothness term weight and δ is Kroneckerfs delta. Based on the data and smoothness terms defined above, the max-flow algorithm gives the globally optimized binary volumetric image $I_{bin}(\mathbf{x}; \alpha)$ (Fig. 4(d)).

Moreover, considering the effect of boundary dithering, boundary pixels are replaced by the blended morphing image to create the final resultant image $I_{morph}(\mathbf{x}; \alpha)$ (Fig. 4(e)) as

$$I_{morph}(\mathbf{x}; \alpha) = \begin{cases} I_{blend}(\mathbf{x}; \alpha) & \mathbf{x} \text{ is inner or outer boundary} \\ I_{bin}(\mathbf{x}; \alpha) & \text{otherwise.} \end{cases} \quad (12)$$

As mentioned in 3.1, FCM clusters are preferable to k-means clustering in terms of denoising. They also tend to create a smoother blended morphing image than k-means clusters, with the result that artifacts such as holes and cracks in the silhouette region become less prominent, and are more easily recovered by graph-cut denoising.

4 Experiments

4.1 Simple Shapes

In these experiments, several morphing examples of shapes with different genera are shown. The first example is the simplest, that is, morphing from a disk to an annulus as shown in Fig. 6.

Starting from the disk ($\alpha = 0.0$), a silhouette hole appears near the center of the disk at $\alpha = 0.2$ and the genus of the shape changes from 0 to 1 at this time. Then, the hole is gradually dilated as the transition parameter increases and the shape coincides with the annulus at the end of the transition ($\alpha = 1.0$). This kind of topological change is unique to the proposed approach.

Although the artifacts are visible in the blended morphing (top row of Fig. 6), they are deleted by the graph-cut denoising (middle row of Fig. 6) and the boundary dithering (bottom row of Fig. 6) provides a visually desirable result.

Additional examples of morphing from a disk to double, triple, and quad annuli are shown in Fig. 7. In a similar manner to the previous example, multiple silhouette holes appear in the early stage of the transition and these are gradually dilated. Note that the number of holes appearing coincides with the genus of the corresponding destination shape.

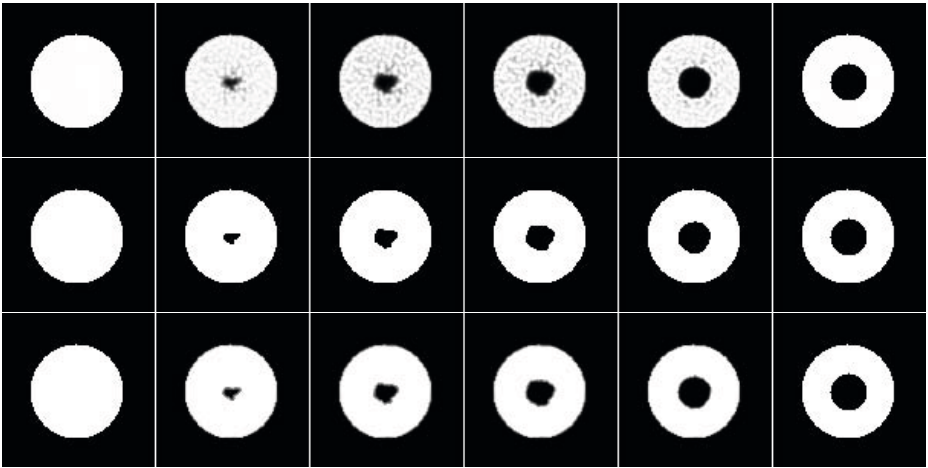


Fig. 6. Morphing from a disk (leftmost) to an annulus (rightmost). The middle four images are morphing images with transition parameters $\alpha = 0.2, 0.4, 0.6,$ and $0.8,$ respectively. Top row: blended morphing, middle row: denoised binary morphing, bottom row: boundary-dithered morphing.

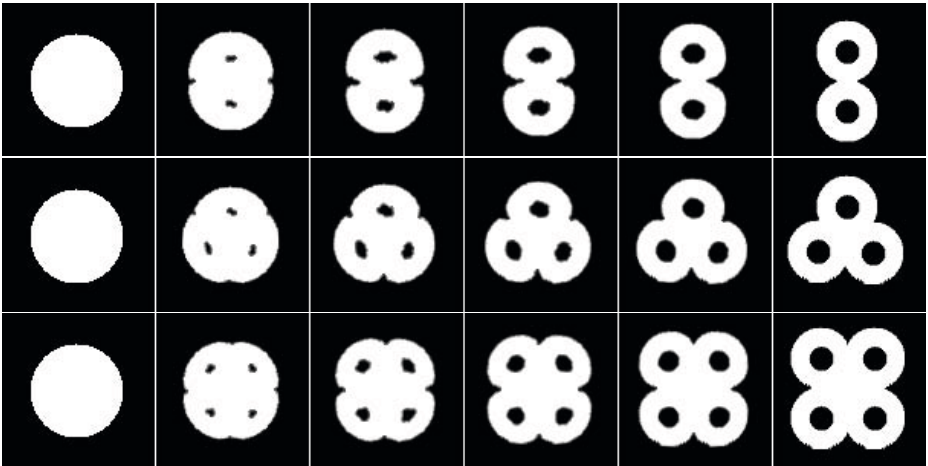


Fig. 7. Morphing from a disk (leftmost) to multiple annuli (rightmost). The middle four images show morphing with transition parameters $\alpha = 0.2, 0.4, 0.6,$ and $0.8,$ respectively. Top row: double annulus, middle row: triple annulus, bottom row: quad annulus.

4.2 Real Shapes

The following examples involve gait silhouette morphs between two postures selected from gait silhouette sequences captured at a 60 fps frame-rate. The source posture includes a silhouette hole in the leg region (leftmost image in

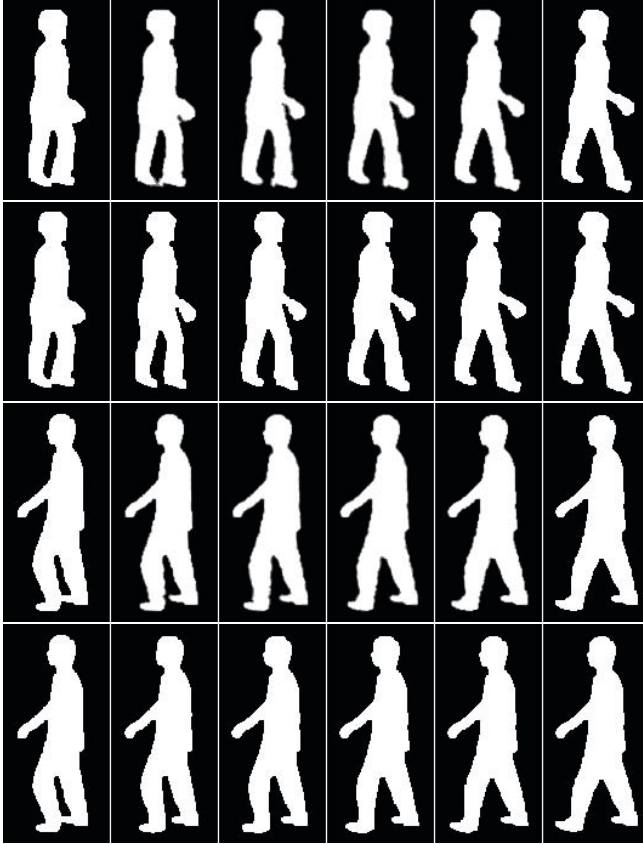


Fig. 8. Morphing from a 1 genus gait silhouette (leftmost) to a 0 genus one (rightmost). First and second rows: boundary-dithered morphing and an original gait sequence, respectively, from an oblique view. Third and fourth rows: the same, but from a side view.

Fig. 8) while the destination shape is expressed as a single closed curve (rightmost image in Fig. 8).

We can see that the resulting morphed images (the first and third rows of Fig. 8) are similar to the original gait sequences between the two postures (the second and fourth rows of Fig. 8). Therefore, the proposed method has real potential for use in many pattern recognition and image processing areas. For example, in a shape matching problem, intermediate shapes of the two key shapes can be generated for the purpose of training sample enhancements even in the presence of topological changes. In addition, when a low frame-rate sequence is provided in an action recognition or gait recognition problem, inter-frame silhouettes can be interpolated and a temporal super-resolution sequence provided for better recognition without worrying about topological changes in the postures.

4.3 Shapes with Large Deformation

The third example shows morphing between alphabetical characters, which involves much larger deformation than in the previous two examples. First, we focus on morphing from "A" to "B". When the squared Euclidean distance d_{jk} is used as the transportation cost t_{jk} , cluster flows around the horizontal middle bar in "A" are directed mainly in two directions: those that go upwards and others that go downwards across a wide NOR region as shown in the first row of Fig. 9. On the other hand, when the NOR region-crossing distance d_{jk}^{NOR} is combined with the transportation cost t_{jk} , all the clusters in the middle bar go upwards across a narrower NOR region than the one in the row below (see the leftmost image in the second row of Fig. 9), as the flows crossing the wide NOR region below are penalized by additional transportation costs. As a result, isolated morphs that appear when using only the squared Euclidean distance are suppressed (see the second row of images in Fig. 9). The morphing from "B" to "C" is also a similar case with that from "A" to "B". While cluster flows of the middle horizontal bar of B go in various directions when using the squared Euclidean distance (the third row of images in Fig. 9), they are limited to three directions when the NOR region-crossing distance is added (the fourth row of images in Fig. 9).

This kind of character morphing can possibly serve as a novel transition effect technique for video editing applications. Compared with existing transitions such

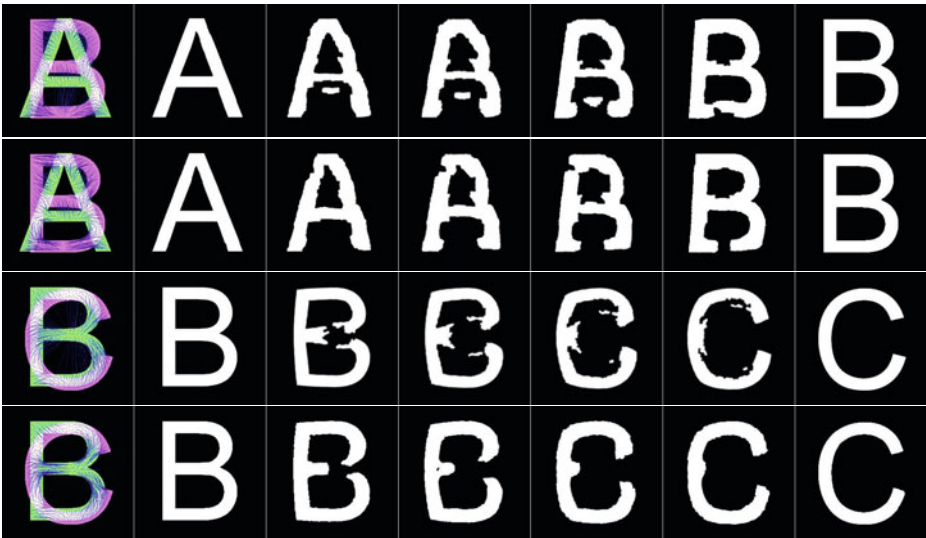


Fig. 9. EMD flows (leftmost) and morphing between alphabetical characters "A" (1 genus), "B" (2 genus) and "C" (0 genus). In EMD flow acquisition process, the squared Euclidean distance is used in each odd-row image, whereas the NOR region-crossing distance is used in conjunction with the squared Euclidean distance in each even-row image.

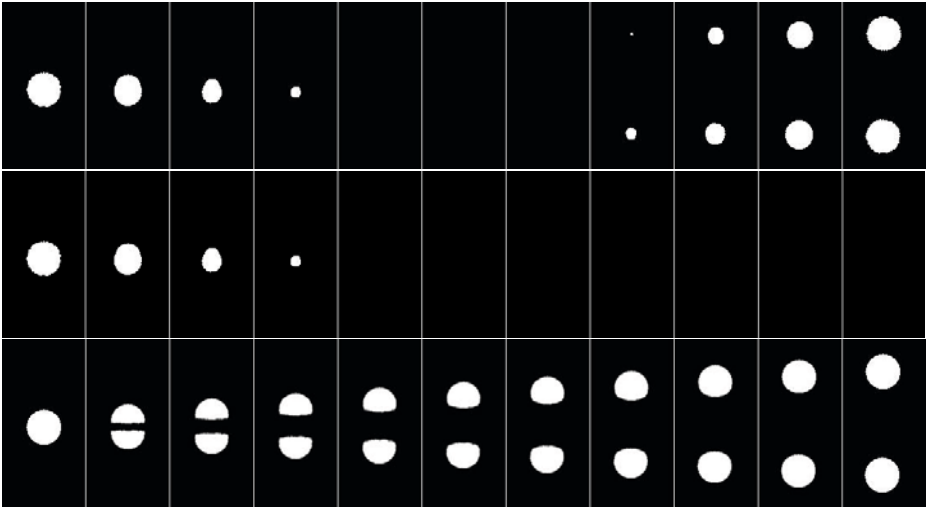


Fig. 10. Morphing from a single disk (leftmost) to two split disks (rightmost). Top row: signed distance field interpolation [23], middle row: narrow-band level-set method [25], bottom row: the proposed method.

as cut, fade-in/out, slide, and wipe, the proposed morphing method provides a unique transition effect. In addition, considering the recent progress in interactive/automatic segmentation techniques [36], not only the characters in a title or caption scene, but also arbitrary objects' silhouettes, can be morphed once they have been extracted by a particular segmentation method. When applying morphing to such textured objects, color transfer on the morph should also be considered in the future.

4.4 Split Disks

Our final example is morphing from a single disk to two split disks as shown in Fig. 10. If the signed distance field interpolation [23] is applied to this example, the source disk is eroded and finally disappears in the transition process, while the two destination disks appear as points in the centers and are dilated to the destination disks. By applying level set-based morphing [25], only the source disk is eroded and finally disappears in the transition process. Unlike these approaches, the proposed method gives a more interesting morphing process where the source disk is initially split into two hemisphere-like shapes, which then move to the destination position by changing their shapes from hemispheres to disks. This kind of morphing process is unique to the proposed method.

5 Discussion

In the proposed morphing process, a many-to-many correspondence of cluster-based EMD flows is used directly for cluster-based morphing. The automatically

obtained correspondence is also provided to construct the existing warping functions [7] [8] [9] [10] [11] [12] between shapes with arbitrary genera.

Another point is that the cluster-based EMD flow can be applied to achieve many purposes, that is, not only morphing, but also shape matching, deformable model construction, and motion analysis without worrying about topological changes, since existing contour-based methods [31] are used within the closed curve/surface shapes. Unlike the optical flows extracted from a textured image sequence that correspond to real motion, the region cluster-based EMD flows do not correspond to real motion, but to *pseudo motion*. This pseudo motion, however, still has potential as a novel motion feature for silhouette-based motion analysis.

On the other hand, the EMD framework allows too much flex flows in several cases (e.g., large deformation of alphabetical characters in [4.3]). We need to introduce additional schemes such as regularity constraints and non-linear interpolation in order to maintain shape well during the morph in the future.

6 Conclusion

This paper described a method for topology-free shape morphing based on region cluster-based EMD flows. First, the region was decomposed into a number of small clusters by FCM clustering and a histogram of the clusters was constructed. Next, the EMD between the two histograms was calculated with the resultant flows and position displacement between the clusters serving as a weighted many-to-many correspondence. A fuzzy cluster-based morphing transition was provided by the obtained correspondence. Finally, the three-dimensional graph-cut binary denoising was applied to reduce artifacts caused by the cluster-based morphing.

Future works are listed below.

- Texture transfer on the morph in conjunction with automatic/interactive segmentation.
- Warping function reconstruction from the weighted many-to-many correspondence based on the EMD flows.
- Application of *pseudo motion* within the shape to silhouette-based recognition, as in action recognition and gait-based person identification.

Acknowledgement. This work was supported by Grant-in-Aid for Scientific Research(S) 21220003.

References

1. Wolberg, G.: Image morphing: a survey. *The Visual Computer* 14, 360–372 (1998)
2. Seitz, S.M., Dyer, C.R.: View morphing. In: *Proc. of ACM SIGGRAP 1996*, pp. 21–30 (1996)
3. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3d faces. In: *Proc. of ACM SIGGRAPH (1999)*

4. Beymer, D., Poggio, T.: Image representations for visual learning. *Science* 272, 1905–1909 (1995)
5. Cootes, T., Taylor, C., Cooper, D., Graham, J.: Active shape models - their training and application. *Computer Vision and Image Understanding* 61, 38–59 (1995)
6. Cootes, T., Edwards, G., Taylor, C.: Active appearance models. *IEEE TPAMI* 23, 681–685 (2001)
7. Wolberg, G.: *Digital image warping*. IEEE Computer Society Press, Los Alamitos (1990)
8. Beier, T., Neely, S.: Feature-based image metamorphosis. *Computer Graphics* 26, 35–42 (1992)
9. Arad, N., Dyn, N., Reifeld, D., Yeshurun, Y.: Image warping by radial basis functions: applications to facial expressions. *CVGIP: Graph Models Image Processing* 56, 161–172 (1994)
10. Lee, S., Chwa, K.Y., Hahn, J., Shin, S.: Image morphing using deformable surfaces. In: *Proc. of Computer Animation 1994*, pp. 31–39. IEEE Computer Society Press, Los Alamitos (1994)
11. Chwa, K.Y., Hahn, J., Shin, S.: Image morphing using deformation techniques. *J. of Visualization Computer Animation* 7, 3–23 (1996)
12. Lee, S.Y., Chwa, K.Y., Shin, S.Y.: Image metamorphosis using snakes and free-form deformations. In: *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1995*, pp. 439–448. ACM, New York (1995)
13. Lee, S., Wolberg, G., Yong Chwa, K., Shin, S.Y.: Image metamorphosis with scattered feature constraints. *IEEE Transactions on Visualization and Computer Graphics* 2, 337–354 (1996)
14. Gao, P., Sederberg, T.W.: A work minimization approach to image morphing. *The Visual Computer* 14, 390–400 (1998)
15. Toshev, A., Shi, J., Daniilidis, K.: Image matching via saliency region correspondences. In: *Proc. of IEEE computer society conference on Computer Vision and Pattern Recognition*, pp. 1–8 (2007)
16. Tekalp, M.: *Digital video processing*. Prentice-Hall, Englewood Cliffs (1995)
17. Chen, S.E., Williams, L.: View interpolation for image synthesis. In: *Proc. of ACM SIGGRAPH*, pp. 279–288 (1993)
18. Shechtman, E., Rav-Acha, A., Irani, M., Seitz, S.M.: Regenerative morphing. In: *Proc. of IEEE computer society conference on Computer Vision and Pattern Recognition 2010, San Francisco, CA, USA*, pp. 1–8 (2010)
19. Sederberg, T.W.: A physically based approach to 2-d shape blending. *Computer Graphics* 26 (1992)
20. Kent, J., Carlson, W., Parent, R.: Shape transformation for polyhedral objects. *Computer Graphics* 26, 47–54 (1992)
21. DeCarlo, D., Gallier, J.: Topological evolution of surfaces. In: *Proc. Graphics Interface*, pp. 194–203 (1996)
22. Fu, H., Tai, C.L., Zhang, H.: Topology-free cut-and-paste editing over meshes. In: *Proceedings of the Geometric Modeling and Processing, GMP 2004*, p. 173. IEEE Computer Society, Washington, DC (2004)
23. Payne, B., Toga, A.: Distance field manipulation of surface models. *IEEE Computer Graphics and Applications* 12, 65–71 (1992)
24. Cohen-Or, D., Levin, D., Solomivici, A.: Three-dimensional distance field metamorphosis. *ACM Trans. Graphics* 17, 116–141 (1998)

25. Breen, D., Whitaker, R.: A level-set approach for the metamorphosis of solid models. *IEEE Transactions on Visualization and Computer Graphics* 7, 172–192 (2001)
26. Castro, G., Ugail, H.: Shape morphing of complex geometries using partial differential equations. *Journal of Multimedia* 2, 15–25 (2007)
27. Osher, S., Sethian, J.: Fronts propagating with curvature dependent speed: Algorithm based on hamilton-jacobi formation. *Journal of Computational Physics* 79, 12–49 (1988)
28. Zhu, L., Yang, Y., Tannenbaum, A., Haker, S.: Image morphing based on mutual information and optimal mass transport. In: *Int. Conf. on Image Processing*, pp. 1675–1678 (2004)
29. Zhu, L., Yang, Y., Haker, S., Tannenbaum, A.: An image morphing technique based on optimal mass preserving mapping. *IEEE Transactions on Image Processing* 16, 1481–1495 (2007)
30. Kaziska, D., Srivastava, A.: Cyclostationary processes on shape spaces for gait-based recognition. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) *ECCV 2006*. LNCS, vol. 3952, pp. 442–453. Springer, Heidelberg (2006)
31. Grauman, K., Darrell, T.: Fast contour matching using approximate earth mover's distance. In: *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pp. 220–227 (2004)
32. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: *Numerical Geometry of Non-Rigid Shapes*. Springer, Heidelberg (2008)
33. Schneider, D.C., Eisert, P.: Fast nonrigid mesh registration with a data-driven deformation prior. In: *Proc. ICCV Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment, NORDIA* (2009)
34. Sagawa, R., Akasaka, K., Yagi, Y., Hamer, H., Gool, L.V.: Elastic convolved icp for the registration of deformable objects. In: *Proc. 2009 IEEE 12th International Conference on Computer Vision Workshops (3DIM 2009)*, Kyoto, Japan, pp. 1558–1565 (2009)
35. Hoppner, F., Klawonn, F., Kruse, R., Runkler, T.: *Fuzzy Cluster Analysis*. John Wiley and Sons, Chichester (1999)
36. Li, Y., Sun, J., Shum, H.Y.: Video object cut and paste. *ACM Trans. Graph.* 24, 595–600 (2005)

Object Detection Using Local Difference Patterns

Satoshi Yoshinaga, Atsushi Shimada,
Hajime Nagahara, and Rin-ichiro Taniguchi

Graduate School of Information Science and Electrical Engineering,
Kyushu University, Fukuoka, Japan
{yoshinaga, atsushi, nagahara, rin}@limu.ait.kyushu-u.ac.jp

Abstract. We propose a new method of background modeling for object detection. Many background models have been previously proposed, and they are divided into two types: “*pixel-based models*” which model stochastic changes in the value of each pixel and “*spatial-based models*” which model a local texture around each pixel. Pixel-based models are effective for periodic changes of pixel values, but they cannot deal with sudden illumination changes. On the contrary, spatial-based models are effective for sudden illumination changes, but they cannot deal with periodic change of pixel values, which often vary the textures. To solve these problems, we propose a new probabilistic background model integrating pixel-based and spatial-based models by considering the illumination fluctuation in localized regions. Several experiments show the effectiveness of our approach.

1 Introduction

Background subtraction is one of the most widely used techniques to detect moving objects from image sequences. It enables us to detect objects by calculating subtraction of a background image from an observed image without any specific prior information about moving objects. However, when we use a simple background image in outdoor scenes, it will detect not only object regions but also a lot of noise regions. This is because it is very sensitive to changes in the pixel values caused by waving trees, fleeting clouds, illumination changes and so on. Therefore, many approaches to model these background changes have been proposed [1–9].

In general, the approaches of background modeling can be divided into two types: “*pixel-based approach*” and “*spatial-based approach*”. In the case of pixel-based background models, they commonly have a probability density function (PDF) for each pixel to represent the pixel value distribution observed in a video sequence. Stauffer et al. proposed a background estimation method, in which mixture-of-Gaussians is used to describe the background model [4], and Shimada et al. augmented this method by introducing a mechanism to change the number of Gaussians dynamically in each pixel [5]. Elgammal et al. employed Parzen density estimation to estimate the PDF of the pixel value non-parametrically

[6]. These pixel-based models are effective for periodic changes of pixel values, which are caused by fleeting clouds, movement of tree branches or leaves, waves on water and so on. However, they cannot adapt for sudden illumination changes. This is because they construct their models based on statistical information of the pixel values observed in the past.

In the case of spatial-based background models, they model local textures in a localized region centered around each pixel to evaluate the similarity between the background image and the observed image [2, 3]. These models define several pairs of a target pixel and its neighbor pixels, and establish a background model using magnitude relations of pixel values of those pairs. Therefore, spatial-based background models are more robust than pixel-based one against sudden illumination changes, because there are little changes in magnitude relations of pixel values before and after a sudden illumination change. On the other hand, they cannot deal with periodic changes of pixel values, which are caused by the movement of tree branches or leaves and so on, since the textures change in such situations.

The *hybrid background models* are also proposed, in which both a pixel-based and spatial-based background models are utilized. Tanaka et al. proposed a hybrid background model [9], in which they combined the results of a pixel-based background model [8] and spatial-based one using “logical AND”. Their model is more robust than pixel-based or spatial-based ones, since it can utilize both properties by combining the results of two different models. However, objects should be detected accurately by both models, and mis-detection in either of the two models reduces the detection accuracy. Therefore, hybrid models require more sophisticated combinatorial algorithm for integrating the results of two different models with high accuracy.

In this paper, we propose a new probabilistic background model by integrating the methodology of both a pixel-based and a spatial-based approaches. Note that our approach, unlike previous works [9], does not combine two approaches in a naive way. We will give a detail explanation about our proposed method in Section 2.

2 Probabilistic Background Model Considering Illumination Fluctuation in Localized Region

We propose a new probabilistic background model, as shown in Fig. 1, by considering the illumination fluctuation in a localized region centered around each pixel.

2.1 Design of Local Difference Pattern

In our background model, the methodologies of both a pixel-based and a spatial-based approaches are naturally integrated. In the case of pixel-based model, the problem is that spatial information (e.g. texture) was not considered. On the

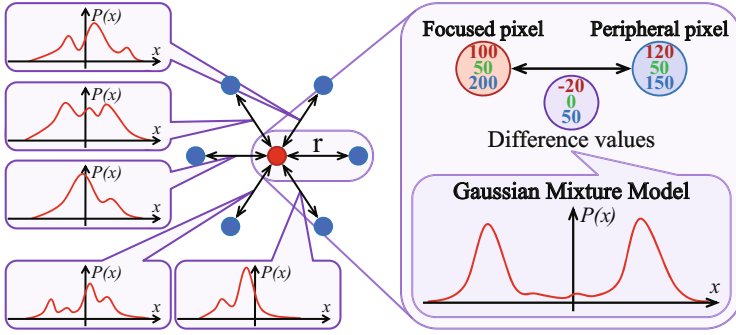


Fig. 1. Probabilistic background model using LDP: Our proposed model defines several pairs of focused pixel and its peripheral pixels in localized region which is a circular region with radius r . Each pair has a Gaussian Mixture Model (GMM) to model the distribution of the difference between the pixel values of them.

contrary, in the case of spatial-based model, local texture was represented in magnitude relations of pixel values in a background image and multiple hypotheses of the background can not be maintained, which causes a problem.

To solve these problems, we propose a new probabilistic background model integrating pixel-based and spatial-based models by considering the illumination fluctuation in localized regions, as shown in Fig. 1. In the proposed model, we define several pairs of a focused pixel and its peripheral pixels, i.e., its surrounding pixels, in a localized region (Fig. 1 is an example where the number of pair is 6), and we give each pair a Gaussian Mixture Model (GMM) to model the distribution of the difference between pixel values of each pair. Here, we call these pixel value differences in the localized region “*Local Difference Pattern*” (LDP).

The advantages of using LDP are as follows (see Fig. 2). In most cases where sudden illumination changes occur, there are little changes in a LDP, since the pixel values in a localized region similarly increase and decrease in their values. Therefore, our proposed method can deal with sudden illumination changes as shown in Fig. 2 (a). Furthermore, our proposed method can also deal with periodic changes of pixel values, since GMM represents multiple hypotheses of the background as shown in Fig. 2 (b). Thus, our background model can utilize both properties of pixel-based and spatial-based model, without decreasing the accuracy.

2.2 Construction of Local Difference Pattern

A focused pixel in an observed image is represented by a vector $\mathbf{p}_c = (x_c, y_c)^T$. A directional vector $\mathbf{a}_j (j = 1, \dots, N_{pair})$, which represents the direction of each reach or the direction of each peripheral pixel, is defined as follows.

$$\mathbf{a}_j = \left(\cos \frac{j-1}{N_{pair}} 2\pi, \sin \frac{j-1}{N_{pair}} 2\pi \right)^T \tag{1}$$

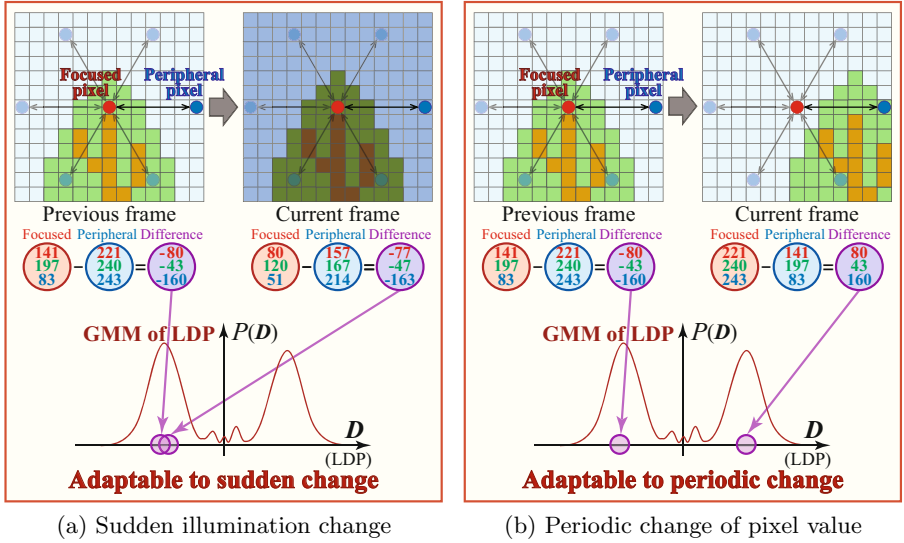


Fig. 2. Adaptivities of the proposed model to background fluctuation: (a) shows the case that illumination suddenly changed (e.g. when sunlight is blocked out by clouds, etc.). LDP can absorb the effect of illumination changes, since it globally affects pixel values as a bias. (b) shows the case that texture is periodically changed (e.g. effect of movement of tree or grass, waves on water, etc.). GMM can also adapt to these kinds of deriodic changes, since it allows for multiple hypotheses of the background.

Each peripheral pixel $\mathbf{p}_j = (x_j, y_j)^T$, which is present on the circumference of a circle with radius r centered around a focused pixel \mathbf{p}_c , is represented by $\mathbf{p}_j = \mathbf{p}_c + r\mathbf{a}_j (j = 1, \dots, N_{pair})$, where N_{pair} is the number of peripheral pixels.

We define N_{pair} pairs of a focused pixel \mathbf{p}_c and its peripheral pixels \mathbf{p}_j . Then, a LDP observed at a focused pixel \mathbf{p}_c at time t is defined by the difference between the pixel values of each pair, and we represent it by $\mathbf{D}^t = \{\mathbf{X}_1^t, \dots, \mathbf{X}_j^t, \dots, \mathbf{X}_{N_{pair}}^t\}$. Here, $\mathbf{X}_j^t = f(\mathbf{p}_c) - f(\mathbf{p}_j)$, where $f(\mathbf{p})$ is the d -dimensional vector representing the value of pixel \mathbf{p} ($d = 3$ in case of RGB color images). Fig. 1 shows an example of $N_{pair} = 6$.

In most cases where sudden illumination changes occur, there is little change in the LDP. Therefore, our proposed method based on the LDP can deal with sudden illumination changes.

2.3 Probabilistic Background Model Based on LDP

In our proposed method, we give each pair a GMM to represent the PDFs of a LDP. Here, we focus on the j -th pair of a LDP, and we model the difference between the pixel values of the j -th pair \mathbf{X}_j^t . Let $\{\mathbf{X}_j^1, \dots, \mathbf{X}_j^t\}$ be the difference between the pixel values of the j -th pair observed until time t , then we can represent a PDF of them by a mixture of K Gaussian distributions. Then, the probability of observing the difference is

$$P(\mathbf{X}_j^t) = \sum_{k=1}^K w_{j,k}^t \eta(\mathbf{X}_j^t | \boldsymbol{\mu}_{j,k}^t, \boldsymbol{\Sigma}_{j,k}^t) \quad (2)$$

where j is a subscript representing the direction of the peripheral pixel based on the focused pixel, $w_{j,k}^t$, $\boldsymbol{\mu}_{j,k}^t$, $\boldsymbol{\Sigma}_{j,k}^t$ are a weight, the mean and the covariance matrix of the k -th Gaussian in the mixture at time t , and η is a Gaussian probability density function as follows.

$$\eta(\mathbf{X}_j^t | \boldsymbol{\mu}_{j,k}^t, \boldsymbol{\Sigma}_{j,k}^t) = \frac{1}{(2\pi)^{\frac{d}{2}} |\boldsymbol{\Sigma}_{j,k}^t|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{X}_j^t - \boldsymbol{\mu}_{j,k}^t)^T \boldsymbol{\Sigma}_{j,k}^{-1} (\mathbf{X}_j^t - \boldsymbol{\mu}_{j,k}^t)\right) \quad (3)$$

K is determined by the available memory and computational power. Also, to reduce the computation cost, the covariance matrix is assumed to be of the form:

$$\boldsymbol{\Sigma}_{j,k}^t = \sigma_{j,k}^t \mathbf{I} \quad (4)$$

In the case of RGB color space, this means that the red, green, and blue pixel values are independent and have the same variances. While this is certainly not the case, the assumption allows us to avoid a costly matrix inversion at the expense of some accuracy.

Thus, a PDF of the difference between pixel values of a pair of LDP observed until time t is characterized by a mixture of K Gaussian distributions. A new difference value will be represented by one of the components of the mixture model and used to update the model. We will describe the background model estimation process in 6 steps.

Step1. Every new difference value \mathbf{X}_j^t is examined against the existing K Gaussian distributions until a match is found. Here, the match is defined as a difference within 2.5 standard deviations of distribution.

Step2. The prior weights $w_{j,k}^t$ of the K distributions of j -th GMM at time t are updated as follows

$$w_{j,k}^t = (1 - \alpha)w_{j,k}^{t-1} + \alpha M_{j,k}^t \quad (5)$$

where α is the learning rate and $M_{j,k}^t$ is 1 for the matched distribution and 0 for the remaining distributions. After this process, these weights $w_{j,k}^t$ are renormalized.

Step3. The $\boldsymbol{\mu}_{j,k}^t$ and $\sigma_{j,k}^t$ parameters for unmatched distributions remain unchanged. The parameters of the distribution which matches the new observation are updated as follows

$$\boldsymbol{\mu}_{j,k}^t = (1 - \rho)\boldsymbol{\mu}_{j,k}^{t-1} + \rho \mathbf{X}_j^t \quad (6)$$

$$\sigma_{j,k}^t = (1 - \rho)\sigma_{j,k}^{t-1} + \rho(\mathbf{X}_j^t - \boldsymbol{\mu}_{j,k}^t)^T (\mathbf{X}_j^t - \boldsymbol{\mu}_{j,k}^t) \quad (7)$$

where ρ is the second learning rate and is defined as follows.

$$\rho = \alpha \eta(\mathbf{X}_j^t | \boldsymbol{\mu}_{j,k}^t, \boldsymbol{\Sigma}_{j,k}^t) \quad (8)$$

Step4. If none of the K distributions match the current difference value in **Step1**, a new Gaussian distribution is created as follows

$$w_{j,K+1}^t = W \quad (9)$$

$$\boldsymbol{\mu}_{j,K+1}^t = \mathbf{X}_j^t \quad (10)$$

$$\sigma_{j,K+1}^t = \sigma_{j,K}^t \quad (11)$$

where W is the initial weight value¹ for the new Gaussian. After this process, the weights are renormalized.

Step4-1: When the weight of the least probable distribution is smaller than a threshold, the distribution is deleted, and the remaining weights are renormalized.

Step4-2: When the difference between means of two Gaussians (the one is η_a and the other is η_b) is smaller than a threshold, these distributions are integrated into one Gaussian. The new weight, mean and variance of integrated Gaussian η_c are calculated as follows.

$$w_{j,c}^t = w_{j,a}^t + w_{j,b}^t \quad (12)$$

$$\boldsymbol{\mu}_{j,c}^t = \frac{w_{j,a}^t \boldsymbol{\mu}_{j,a}^t + w_{j,b}^t \boldsymbol{\mu}_{j,b}^t}{w_{j,a}^t + w_{j,b}^t} \quad (13)$$

$$\sigma_{j,c}^t = \frac{w_{j,a}^t \sigma_{j,a}^t + w_{j,b}^t \sigma_{j,b}^t}{w_{j,a}^t + w_{j,b}^t} \quad (14)$$

Step5. The Gaussians are ordered by the value of w/σ . This value increases as the distribution gains more evidence and as the variance decreases.

Step6. The first B distributions are chosen as the background model, and B is represented as follows

$$B_j = \operatorname{argmin}_b \left(\sum_{k=1}^b w_{j,k}^t > T \right) \quad (15)$$

where T is a measure of the minimum portion of the data that should be accounted for by the background. If a small value for T is chosen, the background model is usually unimodal. If T is higher, a multi-modal distribution is created by repetitive background changes (e.g. fleeting clouds, the movement of tree branches or leaves, the waves on water, etc.) could result in multiple colors being included in the background model. This results in a transparency effect which allows the background to accept two or more separate colors.

¹ If W is higher, the distribution is chosen as the background model for a long time.

2.4 Object Detection Using LDP

Object detection based on the LDP using N_{pair} GMMs is defined by following equation.

$$f(x, y) = \begin{cases} background & \text{if } \sum_{j=1}^{N_{pair}} \phi(\mathbf{X}_j^t) > th \\ foreground & \text{otherwise} \end{cases} \quad (16)$$

In Equation 16, $\phi(\mathbf{X}_j^t)$ is a function which returns 1 or 0, according to whether the matched distribution found in **Step1** is one of the background models (described in **Step6**) or not. In addition, th is a threshold for determining whether a focused pixel p_c belongs to the background or the foreground.

3 Experimental Result

We conducted two kinds of experiments. First, we examined the parameters (r, N_{pair}) of LDP and decided one of the good parameters which was used in the following experiment. Second, we compared the accuracy with state-of-the-art methods. Due to space limitation, we'll report the result of PETS2001² dataset.

3.1 Preliminary Experiment for Adjusting Parameters

In our proposed method, we focus on a localized circular region of radius r centered around each pixel, and model a LDP using N_{pair} GMMs. Therefore, we investigated the parameters (r, N_{pair}) of LDP by several experimental analyses. Fig. 3 shows that the variation of the accuracy across the parameters. Here, we employed Recall and Precision for the accuracy, and used manually-produced Ground Truth³ dataset to evaluate them.

Fig. 3 shows that there were little changes in the accuracy, when the number of pair N_{pair} was more than 4. Also it shows that the change of radius r caused little or no change in the accuracy, when N_{pair} was more than 4.

We evaluated the computational cost. The image size was 320×240 (pixel) and the PC had Core 2 Duo 2.8 GHz CPU and 4GB memory. Table 1 shows that computational cost increases in proportion to N_{pair} . Therefore, we have employed $N_{pair} = 6$ and $r = 10$ as an optimal parameter for PETS2001 dataset in terms of the balancing point of the accuracy and the computational cost.

We carried out preliminary experiments for using other data sets and confirmed that N_{pair} is not so changed depending on video contents. Therefore, N_{pair} was not critical for the performance. On the other hand, the radius r depends on the video content. However, it is easy to decide the parameter r from

² Benchmark data of International Workshop on Performance Evaluation of Tracking and Surveillance. Available from <ftp://pets.rdg.ac.uk/PETS2001/>

³ The Ground Truth image denotes foreground regions which should be detected by background subtraction. We made Ground Truth for some Benchmark data including PETS2001 manually, and have published them to the web. Available from <http://limu.ait.kyushu-u.ac.jp/dataset/>

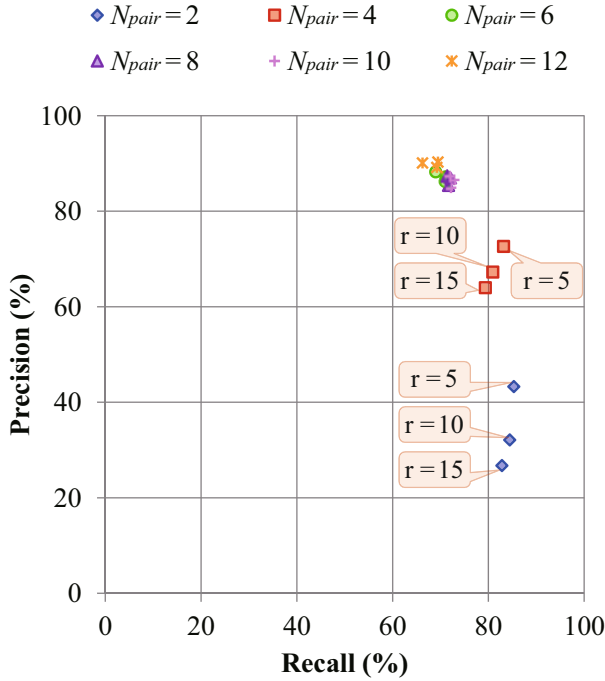


Fig. 3. The accuracy of object detection in relation to the parameters of LDP

a prior knowledge of the object sizes. In most cases (e.g. surveillance, security, etc.), we would predict the size of the objects, since the camera is stationary and observes similar objects in these applications. Hence, it does not lose a generality or effectiveness of the proposed method, although the parameter can be determined by the prior knowledge.

3.2 Object Detection Accuracy

We evaluated the accuracy of object detection based on Recall and Precision. According to experimental result in Section 3.1, the parameters of LDP were

Table 1. Computational cost in relation to the parameter N_{pair}

Parameter	Average processing time (ms)
$N_{pair} = 2$	68.4
$N_{pair} = 4$	149.6
$N_{pair} = 6$	231.1
$N_{pair} = 8$	321.7
$N_{pair} = 10$	391.4
$N_{pair} = 12$	472.3

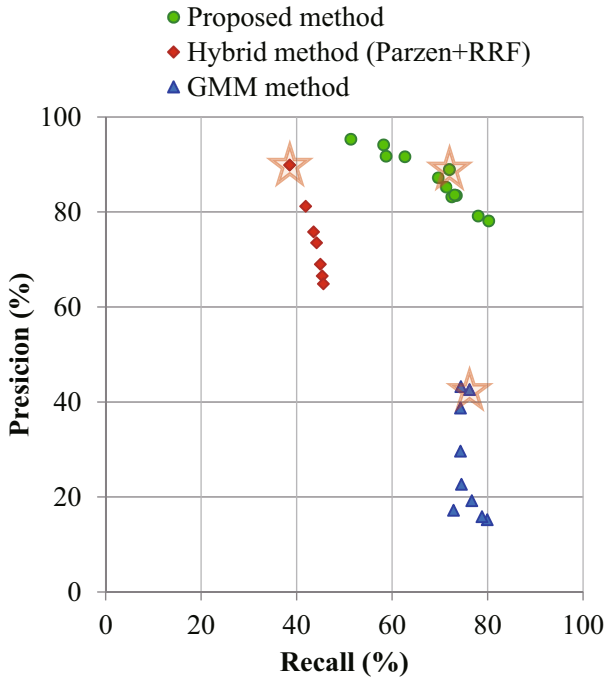


Fig. 4. The accuracy of object detection in relation to the changes of several parameters in each method: Star signs indicate the best performances on each method

fixed $N_{pair} = 6$ and $r = 10$ respectively. However, our approach has some other parameters; the parameters of GMM (e.g. the initial weight W , threshold T , etc.), which affect the accuracy. Thus, we conducted some experiments with varying the parameters. We used GMM method [5] and Hybrid method (Parzen + RRF) [9] to compare the accuracy with our proposed method. These methods had also several parameters, therefore we performed some experiments with varying the parameters of them as well as our proposed method. Fig.4 and Table 2 show the accuracy of each method. The representative results of background subtraction are shown in Fig.5. In Table 2 and Fig.5, the states flagged with a star sign in Fig.4 have been used as the parameters of each method.

We can see that Precision of the Hybrid method was high but Recall of it was low. This was also confirmed from Fig.5 since there was little noise but

Table 2. The accuracy of object detection

	Recall	Precision
Proposed method	72.0%	88.9%
Hybrid method (Parzen + RRF) [9]	38.6%	89.9%
GMM method [5]	76.3%	42.6%

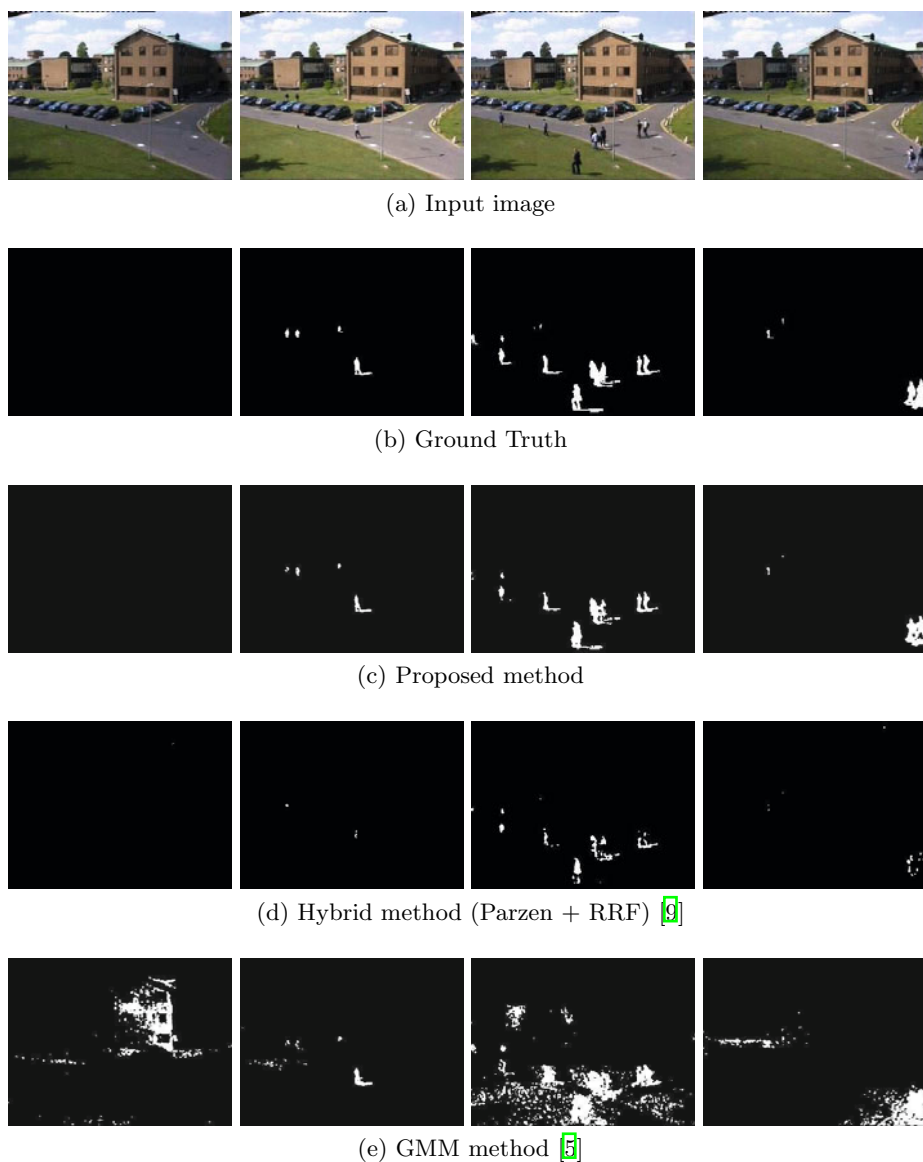


Fig. 5. Object detection by the Proposed method, Hybrid method and GMM method

object regions detected by the Hybrid method were abnormally-small. Fig. 4 and Table 2 also show that Recall of the GMM method was high but Precision of it was low, and the GMM method detected not only objects but also many noises in Fig. 5. On the other hand, both Recall and Precision of our proposed method were high, and object regions were accurately detected with little noise.

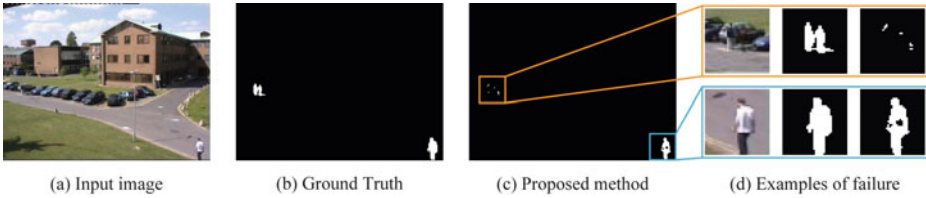


Fig. 6. Problem with Our Proposed Method

3.3 Discussion

Totally, LDP gave use better results than state-of-the-art methods. However, we found out that following problems were caused by the characteristic of LDP that it used the difference values between focused pixel and peripheral pixels.

- **The objects with uniform texture:**

We assumed that the LDP ignores global changes such as illumination changes. However, the LDP sometimes causes a problem in the case where “a object with uniform texture” appears on “the background with uniform texture”. In this case, if radius r is smaller than foreground object, the local changes caused by the objects are mistakenly regarded as global changes. As the result, our proposed model fails to detect internal regions of the objects and False Negative increases (see blue-rectangle in Fig.6).

As discussed in Section3.1, this problem can be avoided by selecting a suitable radius r from a prior knowledge of a scene. It would be desirable to decide the parameter automatically, which will be our future work.

- **Color similarity between object and background:**

In the case where “an object has similar color with background” appears, the difference between the object and background becomes smaller. It is difficult to detect such objects in our method (see orange-rectangle in Fig.6).

4 Conclusion

In this paper, we have proposed a new probabilistic background model using several GMMs. We considered the illumination fluctuation in the localized region, and model LDP (the difference values between the values of focused pixel and its peripheral pixels, which is present on the circumference of circle centered around focused pixel) using GMMs. We could integrate pixel-based and spatial-based model themselves by using LDP, and background model using LDP could utilize both properties without decreasing the accuracy, unlike traditional model. In our experiment, we have got a good result where both Precision and Recall were superior to the traditional background subtraction methods.

Future works are summarized as follows.

- **Reduction of computational time**

Our proposed method have the N_{pair} GMMs, therefore it is cost to update them, where N_{pair} is the number of peripheral pixels. In the case of

Image Size = 320×240 (pixel) and $N_{pair} = 6$, computational time was about $230ms$ using a PC with a Core 2 Duo 2.8GHz CPU and 4GB memory. It is not good for real-time processing, and therefore we should develop a mechanism to reduce the computational time.

– **Improvement of the accuracy of object detection**

Our proposed method has some problems associated with objects, as described in section [3.3](#). Therefore, it is necessary to sophisticate our proposed method to cope with the objects described in section [3.3](#).

References

1. Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: Principle and Practice of Background Maintenance. In: International Conference on Computer Vision, pp. 255–261 (1999)
2. Marko, H., Matti, P.: A Texture-Based Method for Modeling the Background and Detecting Moving Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28, 657–662 (2006)
3. Satoh, Y., Kaneko, S., Niwa, Y., Yamamoto, K.: Robust object detection using a Radial Reach Filter (RRF). *Systems and Computers in Japan* 35, 63–73 (2004)
4. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 246–252 (1999)
5. Shimada, A., Arita, D., Ichiro Taniguchi, R.: Dynamic Control of Adaptive Mixture-of-Gaussians Background Model. In: *CD-ROM Proceedings of IEEE International Conference on Advanced Video and Signal Based Surveillance* (2006)
6. Elgammal, A., Duraiswami, R., Harwood, D., Davis, L.: Background and Foreground Modeling using Non-parametric Kernel Density Estimation for Visual Surveillance. *Proceedings of the IEEE* 90, 1151–1163 (2002)
7. Monari, E., Pasqual, C.: Fusion of Background Estimation Approaches for Motion Detection in Non-static Backgrounds. In: *CD-ROM Proceedings of IEEE International Conference on Advanced Video and Signal Based Surveillance* (2007)
8. Tanaka, T., Shimada, A., Arita, D., Ichiro Taniguchi, R.: A Fast Algorithm for Adaptive Background Model Construction Using Parzen Density Estimation. In: *CD-ROM Proceedings of IEEE International Conference on Advanced Video and Signal Based Surveillance* (2007)
9. Tanaka, T., Shimada, A., Ichiro Taniguchi, R., Yamashita, T., Arita, D.: Towards robust object detection: integrated background modeling based on spatio-temporal features. In: *The Ninth Asian Conference on Computer Vision* (2009)

Randomised Manifold Forests for Principal Angle-Based Face Recognition

Ujwal D. Bonde¹, Tae-Kyun Kim², and K.R. Ramakrishnan¹

¹ Department of Electrical Engg., Indian Institute of Science, Bangalore, India

² Department of Engineering, University of Cambridge, Cambridge, UK

Abstract. In set-based face recognition, each set of face images is often represented as a linear/nonlinear manifold and the Principal Angles (PA) or Kernel PAs are exploited to measure the (dis-)similarity between manifolds. This work systemically evaluates the effect of using different face image representations and different types of kernels in the KPA setup and presents a novel way of randomised learning of manifolds for set-based face recognition. First, our experiments show that sparse features such as Local Binary Patterns and Gabor wavelets significantly improve the accuracy of PA methods over 'pixel intensity'. Combining different features and types of kernels at their best hyper-parameters in a multiple classifier system has further yielded the improved accuracy. Based on the encouraging results, we propose a way of randomised learning of kernel types and hyper-parameters by the set-based Randomised Decision Forests. We observed that the proposed method with linear kernels efficiently competes with those of nonlinear kernels. Further incorporation of discriminative information by constrained subspaces in the proposed method has effectively improved the accuracy. In the experiments over the challenging data sets, the proposed methods improve the accuracy of the standard KPA method by about 35 percent and outperform the Support Vector Machine with the set-kernels manually tuned.

1 Introduction

Unlike traditional access control scenarios, face recognition in dynamic environments is yet extremely challenging due to uncontrolled lighting conditions, large pose variations, facial expressions and severe occlusions. For the past decade set-based face recognition has gained a huge interest in related fields. Over conventional single-shot based face recognition, the main benefits have been two folds: a) its ability to represent and match data over a combination of face exemplars and b) its natural extension to videos where a tracked object can be represented as a set of images. This has led to significant improvement in accuracy and efficiency for face recognition.

Among different methods for set-based face recognition, the most widespread one is the Principal Angle (PA) method. It represents a set of face images as a subspace and matches one set to another set using subspace angles. Despite the popularity of the Principal Angle based methods, it has not received much

attention on its efficacy using state-of-the-art face image representations (e.g. Local Binary Patterns, Gabor features) other than 'pixel intensity': one reason for this could be that their very sparse representations might be thought difficult to be constrained on linear subspaces. Nonlinear extension of the Principal Angle method by a kernel trick [26] or a set of linear subspaces [15,31] and discriminative versions of the PA technique e.g. Constrained Mutual Subspace Method [8] have been successfully developed. They have shown significantly improved accuracy over the standard method but their good performance is highly subjective to the settings in the methods. In the Kernel Principal Angle technique (KPA) [26], it is not a trivial problem to automatically set the best types of kernels and kernel hyper-parameters. This paper systematically evaluates the Principal Angle methods over a number of respective issues and proposes a novel way of randomised learning for the PA methods using Randomised Decision Forests [2,3]. In this work we look at the following key areas in the framework of Principal Angle based face recognition:

- Performance of features such as Local Binary Patterns (LBP) and Gabor wavelets, both are sparse representations, over the pixel intensity representation.
- Combining different features and kernels for the KPA method by a multiple classifier system.
- Proposing randomised manifold (or kernel) learning by Random Decision Forests.
- Using the idea of CMSM to incorporate discriminative information for the proposed method.

We demonstrate these for a video-based face recognition problem.

Rest of the paper is structured as follows. In Section 2 we briefly review related work. Section 3 details KPA using non-linear feature extraction techniques and LBPs, followed by the method for combining these features with different kernels as a multiple classifier system. The Random Manifold Forest is proposed in Section 4. Experimental setup and the results obtained are presented in Section 5. We conclude our work in Section 6.

2 Related Work

Use of the principal angles(PA) for matching sets of face images was initially proposed by Yamaguchi *et al.* [28]. This has become more widely applicable since the Kernelized version was proposed by Wolf *et al.* [26]. A large number of related methods including Boosted Manifold Principal Angles [15], Constrained Mutual Subspace Method(CMSM) [8] and Orthogonal Subspace Method(OSM) [9] have been proposed as an improvement over the original PA or KPA method. The PA-based methods have shown superior to other alternatives such as parametric distribution matching and simple aggregation of individual sample matching consistently in literature e.g. [32,31]. However all of the PA methods above have considered raw pixel intensity images for their input and have not paid much attention to representations.

Nonlinear extension of the PA methods has been obtained largely either by a kernel technique [26] or by expressing a manifold as a set of linear subspaces [15, 31]. Although they have been shown better than the standard PA method, their good performance is highly dependent on how to form a nonlinear subspace or manifold. Despite the popularity of the KPA for face recognition, it has not received attention on its effectiveness using different kernels and hyper-parameters. Based on the encouraging results by the LBP and Gabor features, we investigate a way to combine different features and different types of kernels in a multiple classifier system, firstly assuming the best hyper-parameters given, and later propose a novel method of randomised learning for both *kernel types* and their *hyper-parameters* by the KPA and random decision forests [2].

CMSM as a discriminative method has been shown to significantly improve standard PA and KPA techniques. It has since then been used for automatic character listing in videos [14], recognition in image sets [8]. The main drawback in CMSM is the choice of the *sum space* and its dimensionality. Methods such as multiple CMSM [29] and boosted CMSM [30] have been proposed to address this problem to a certain extent. But its efficacy is still restricted to a certain range that needs to be estimated. Here we propose a method that circumvents this question similar to the problem of choosing a kernel and its hyper-parameter in the PA or KPA methods.

Multiple classifier system refers to techniques to aggregate the evidences from multiple sources (or classifiers) and typically provides better performance than individual base classifiers. These techniques have been widely used in combining results obtained in biometric systems and also in face recognition example [11, 16]. A large number of methods have been developed for combining the classifier outputs at different levels: the simplest yet robust methods are the sum and the product rules which combine the classifiers at the measurement/or confidence level [16]. These methods assume that individual classifiers are uncorrelated. Some methods fuse classifiers at the classifier confidence level, establishing the classifier weights by their performances on a validation set [6]. Mixture of experts (MoE) [10] jointly learns multiple classifiers, their weights and data partitions for binary class problems. This was extended to multi-class problems in Chen *et al.* [5]. MoE provides an unified framework of multiple classifier learning and fusion, though it resorts to a local optimal solution due to the iterative algorithm, EM used for optimisation. We have formulated a novel closed-form solution for learning classifier weights for *multi-class* problems and have shown that this outperforms the sum, product, minimum score, maximum score and weighted sum rules, where the weights were set according to the classifier accuracies.

Random Decision Forests (RF) introduced by Breimen [2] and Geurts *et al.* [3] is a powerful ensemble learning technique and has been used in various applications such as image segmentation [23], classification [1] and tracking problems and has shown competitive results in these areas. It is inherently for multi-classes and shows fast learning and classification performance. Randomised learning is useful particularly when features to be learnt are difficult to be explicitly

represented due to a high dimensional space. Comparative studies have been performed on the accuracy of RF against Support Vector Machines (SVM) [24] and in many cases, for example in Gene selection [7], RF was shown superior to the traditional SVMs. In this paper, a novel method for randomised kernel learning is proposed by RF and in the process, Random Manifolds are defined. In the experiments over the challenging data sets, the proposed methods have been shown to outperform the Support Vector Machine with the set-kernels manually tuned.

3 Combining Features and Kernels for KPA

3.1 Base Classifier Design

Previous face recognition methods based on PA have used raw pixel values as features. We use LBP (and Gabor) features which have gained an increasing interest owing to its good performance for classification. Despite having sparse representations these features have shown to perform very well in our setup and have significantly improved the accuracy over the existing methods using raw pixels. In addition, we consider nonlinear feature extraction in KPA before computing the principal angles. In the previous KPA method [26] the dimension of the set subspace is fixed as the set cardinality. However, the intrinsic dimension of the subspace by the set is in general much lower than the number of data points for faces as shown by Kriegman *et al.* [20]. We apply the Kernel PCA technique [22] to get ' k ' (\ll cardinality of set) dimensional approximation of the original subspace before calculating the principal angles. We get the eigenvectors associated with the k largest eigenvalues as $Q_{k(\Phi(A))}, Q_{k(\Phi(B))}$ for the reduced dimensional subspaces. The principal angles between two reduced subspaces are then computed using a kernel trick [26]. Each base classifier is defined as Nearest Neighbor (NN) classifier in terms of the KPA similarity as

$$d(A, B) = \frac{1}{k} \sum_{i=1}^k \cos \theta_i \tag{1}$$

where A, B are two sets of the LBP (or Gabor) vectors and $\cos \theta_i, i = 1, \dots, k$ are the kernel principal angles for the reduced dimensional subspaces and the kernel used. Here all k principal angles are equally considered and feature selection for better recognition accuracy [15] is left as future work. The two features, LBP [27] and Gabor, and three different kernels are used: Gaussian kernel is defined as $K(x, y) = \exp(-\frac{\|x-y\|_2^2}{\sigma^2})$, Fractional power kernel as $K(x, y) = (sign(x^T y) \times (x^T y))^a, 0 < a < 1$, and Polynomial kernel as $K(x, y) = (x^T y)^b, b \geq 1$ respectively.

3.2 Combining Features and Kernels

We propose a novel way of learning classifier weights in multiple classifier system by a closed-form solution. We later show in experiments that this technique

outperforms some baseline methods. The classifiers obtained using two different features (LBP and Gabor) and three different kernels are considered giving a total of six base classifiers.

Let, $\hat{y}_i^c \in \mathbb{R}^Z$ be an indicator vector representing the predicted output of the c -th classifier for the i -th sample with one in the predicted class and zeros elsewhere, $y_i \in \mathbb{R}^Z$ is an indicator vector for the true class label where Z is the number of classes. The cost F is defined by:

$$F = \min_{w^c} \sum_i \left\| \sum_c (w^c \hat{y}_i^c) - y_i \right\|_2^2. \tag{2}$$

Also, if $\hat{Y}_i = [\hat{y}_i^1, \hat{y}_i^2, \dots, \hat{y}_i^C] \in \mathbb{R}^{Z \times C}$ and $w = [w^1, w^2, \dots, w^C]^T \in \mathbb{R}^{C \times 1}$, where C is the number of classifiers (here six), then the cost is rewritten as:

$$F = \min_w \sum_i \|\hat{Y}_i w - y_i\|_2^2.$$

Now if $\hat{Y} = [\hat{Y}_1^T, \hat{Y}_2^T, \dots, \hat{Y}_N^T]^T \in \mathbb{R}^{NZ \times C}$ and $Y = [y_1^T, y_2^T, \dots, y_N^T]^T \in \mathbb{R}^{NZ \times 1}$ where N is the number of data points (or data sets), then the cost function can be further rewritten as: $F = \min_w \|\hat{Y} w - Y\|_2^2 \Rightarrow F = \min_w (w^T \hat{Y}^T \hat{Y} w - w^T \hat{Y}^T Y - Y^T \hat{Y} w - Y^T Y)$. Differentiating it with respect to w and equating it to zero gives:

$$\Rightarrow w_F = (\hat{Y}^T \hat{Y})^{-1} \hat{Y}^T Y. \tag{3}$$

Thus a least square solution for w is obtained using the cost function F .

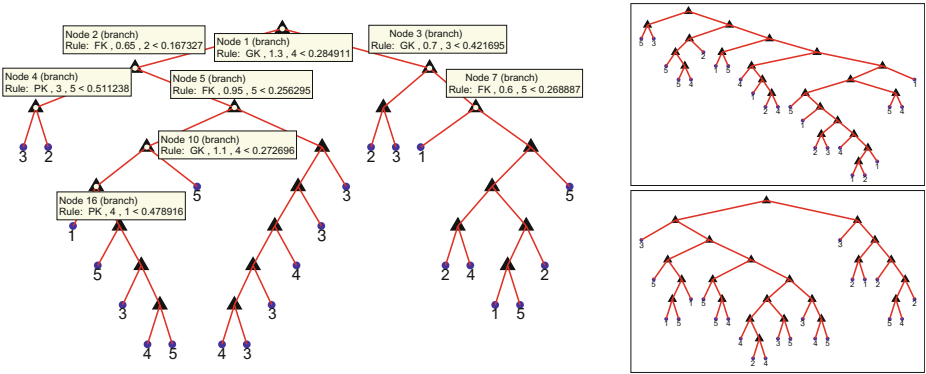


Fig. 1. Example trees in random manifold forests. (left) Decision Rule: First two characters represent the type of kernel ('GK': Gaussian, 'PK': Polynomial, 'FK': Fractional) this is followed by the hyper-parameters of the kernel, next is the reference set followed by the threshold. (right) More example trees in the forests.

4 Random Manifold Forests

Until this section we have not discussed how to obtain the hidden parameters like the hyper-parameters used by the kernels in KPA or the constraint subspace dimension in CMSM. In this section we propose the set based Random Forest method and define Random Manifolds for randomised kernel learning. We have also tested the proposed method with linear kernels and have observed it to efficiently compete with nonlinear kernels. Encouraged by this we go on to propose a method to incorporate discriminative information.

4.1 Random Manifold Forests for Randomised Kernel Learning

We begin by considering a reference set R for each class. At every node we randomly select the following: 1) a kernel type, 2) kernel hyper-parameters and 3) a reference set. The split function at a node for a data set X is defined by

$$f(X, R) = d(X, R) - t = \frac{1}{k} \sum_{i=1}^k \cos \theta_i - t, \tag{4}$$

where k is the reduced subspace dimension, R is the reference set randomly chosen at that node and d is the sum of kernel principal angles for the chosen type of kernel and its hyper-parameters. Note that a set of vectors is taken as input of the split function, whereas a single vector is the input in traditional RFs. This choice is repeated m times from which we select the one that gives us the best split in terms of the information gain [3] [2]. Figure 1 shows the example trees built using this method. The decision taken at few of the nodes for one of the trees is also displayed. At every node we observe that the tree is projecting the data sets to a different feature space depending upon the choice of the kernel and its hyper-parameters. This feature space is split into two regions based on the choice of the reference set R and the threshold t calculated at that node. A decision is taken based on the region in which the subspace spanned by the test data set X lies. The decision surface is given by

$$\mathbb{L} : f(X, R) = 0 \tag{5}$$

It is the separating region at that node. Based on this region the sets X will either go to the left or the right child of the node, i.e:

$$\begin{aligned} I_l &= \{i | f(X_i, R) < 0\} \\ I_r &= I_n \setminus I_l \end{aligned} \tag{6}$$

where I_n is the total data sets arriving at the node n .

For a better intuition let us consider that at a particular node in a tree the sets are projected into a three dimensional feature space and the reference set spans a line as shown in Figure 2 then, the split function and the separating region are given by

$$f(X, R) = \cos \theta_1 - t.$$

$$\mathbb{L} : f(X, R) = 0 \Rightarrow \cos \theta_1 = t. \tag{7}$$

i.e, the threshold t parameterizes a cone as a separating surface. All the sets which span a line (plane) that lies in (passes through) this cone go to the left child and the rest to the right child of this node. This goes on until we reach a leaf node. Thus in essence a leaf node signifies a group of such discriminating regions lying in different spaces. As a result we are no more concerned about choosing a kernel and obtaining its best hyper-parameters since this method allows us to obtain a combination of discriminating regions lying in different spaces. We call this region a *Random Manifold*. Based on this node split strategy, we choose best split functions that maximize the information gain by Shannon entropy [2].

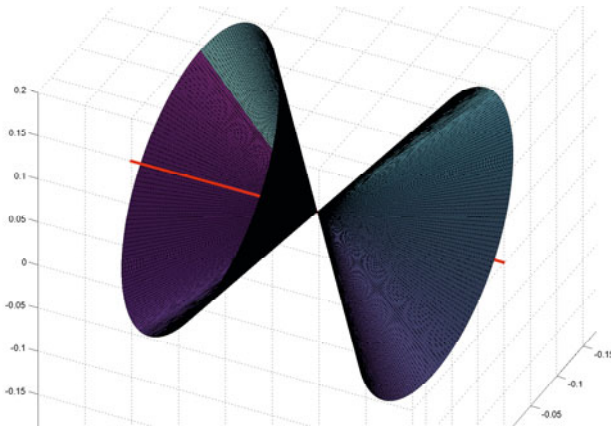


Fig. 2. A three dimensional example for a separating surface. Note that, unlike standard Random Decision Trees, we use set-similarity for splitting nodes.

More Randomness. Inspired from the work on random sampling [25] we also consider using random face subspace for the reference sets, i.e instead of choosing the best k rank approximation, as explained in Section 3.1, we choose the best $k/2$ rank approximates and randomly choose the other $k/2$ bases from the rest of the columns in $Q_{\Phi(X)}$.

In order to further increase the diversity of individual trees in a forests we consider another setting for RF. In this, the threshold t in equation (4) is set randomly rather than being chosen optimally in terms of the information gain. Thus at every node the following need to be randomly chosen: a kernel type, its hyper-parameter, a reference set, its random subspace and the threshold. At each node from all the possible combinations we choose m combinations and retain the one that gives us the best split. This setting is denoted as the Randomised Threshold Random Forest (RtRF).

4.2 Constrained Random Manifold Forests

We have observed that the linear kernels in tree structures were able to capture the inherent non-linearity of the data and gave competitive results compared

with non-linear kernels This motivated us to further incorporate discriminative information obtained from Constrained Mutual subspace matching. CMSM uses only that information which is essential for recognition. However the main problem in CMSM is obtaining the optimal constraint space and its dimension. Similar to our approach in randomised kernel learning we use Random Manifold Forest to learn this. At every node we randomly choose the following parameters: 1) a fixed number of sets per subject from the training data to build the constraint space 2) the dimension of the constraint subspace. As a result of this, similar to the previous setup, during classification at every node, the test set is projected onto a different constraint subspace. The combination of discriminating regions at each leaf node that form the Random Manifolds thus lie in these constrained subspaces.



Fig. 3. Large variations in pose, illumination, expression and scale for a subject in the database. Red outlines show the detected/tracked faces.

More Flexibility. In order to increase the flexibility of random manifolds, instead of a single reference set per subject, we use multiple reference sets. Each of these are obtained by randomly choosing different combinations of the training data.

5 Experimental Results

Experiments were performed in a video based face recognition framework. We have built our own database¹. This database contains 10 subjects having 35 tracks (face sets) which were taken from two sitcoms ‘Coupling’ and ‘Two and a half Men’. These tracks contain anywhere between 10-350 face images in them. Tracks were obtained using a slightly modified version of Anoop *et al*’s tracker [21]. Detector used in this is the Viola Jones detector which detects frontal, left and right profile faces, if the detection is missed then a tracker is initiated to locate the face. Thus apart from the cropped location of the face, the detector/tracker also outputs the state of the face, i.e pose of the detected face (left profile, frontal, right profile) or a tracked face. This additional information was used to build separate manifolds for each detected pose and matching is done only within each these poses. Final similarity is given as the average of the measures obtained from each pose. Large variations in facial poses, illuminations, expressions and backgrounds contained in the database are shown for one of the subjects in Figure 3. The detected/tracked faces are shown by rectangles. Detected faces were resized to 100×100 and passed through the Multi-Scale Retinex filters [12] for compensating illumination changes. See Figure 4

¹ Database will be made available on requesting the author.

Table 1. Performance of different features using Gaussian kernel and Polynomial kernel and CMSM. Results of Fractional Powered kernel are not given due to space constraint. N^* is the number of images in the set.

No. of Training Images	Feature	Gaussian Kernel						Polynomial Kernel						CMSM
		k = N*	k = 50	k = 25	k = 15	k = 10	k = 5	k = N*	k = 50	k = 25	k = 15	k = 10	k = 5	
750	LBP	91.6	94.4	94.8	94.3	93.2	88.5	94.0	95.4	95.3	95.4	95.4	94.6	98.7
	Gabor	91.4	93.8	93.7	93.5	93.3	92.9	93.9	93.4	94.0	94.2	94.1	93.2	97.7
	Raw	72.3	86.2	85.1	82.9	80.8	71.6	74.2	90.7	90.6	88.9	86.7	78.2	95.7
500	LBP	87.6	91.6	91.8	91.5	90.3	85.2	91.7	93.6	94.2	94.3	93.8	92.7	95.1
	Gabor	88.9	90.9	91.1	91.0	90.9	90.2	91.2	91.8	91.6	91.2	90.8	89.9	88.1
	Raw	65.1	82.3	81.2	79.5	76.9	68.9	70.0	85.7	85.8	84.7	82.6	74.7	87.8
250	LBP	76.6	80.6	82.9	83.4	83.5	80.9	81.7	84.4	85.4	86.3	86.4	85.4	86.7
	Gabor	80.3	81.9	82.6	82.9	82.7	81.5	85.3	85.5	85.9	86.0	85.3	84.8	79.7
	Raw	56.9	71.4	75.6	75.5	74.2	70.0	59.0	73.6	78.0	77.9	77.5	72.8	75.3

To further validate our claims we perform experiments using the Oxford database. This database contains detected face of characters from the movies ‘Player’ and ‘Groundhog Day’. We considered 6 subjects that had at least 100 images. We divided these images equally into 10 sets. We used only raw intensity features with a single manifold for all poses.



Fig. 4. Example of normalised face images in the sets

5.1 Performance of KPA and CMSM with Different Features and Kernels

For this experiment only 5 subjects were used from the sitcom database. Three different features, raw pixel intensity, Gabor and LBP were used. As in [27], the LBP feature vectors were set to have the length of 9440 and the Gabor feature vectors the length 9000. For raw pixel we resized the image to 15×15 and raster-scanned it to form a vector of size 225. Results are reported for the three kernels i.e Gaussian, Polynomial and Fractional powered kernels whose best performing kernel hyper-parameters were set with respect to the test set. For the training images (reference sets), tracks (face sets) of each subject were randomly selected so as to have a fixed number of images (750,500 and 250). These images were considered as a single set and were matched against the remaining sets. Each feature and the corresponding kernel projects the faces to a different feature space. The face subspace dimension (described in Section 3.1) at which it performs best needs not be the same. For this reason, we examined

various k values to get the best result. Table 1 shows the accuracies in percentage averaged over 15 different training/testing splits. For CMSM a fixed subspace dimension(30) was considered. LBP and Gabor significantly outperformed the raw pixel features in both KPA and CMSM setup. LBP performed slightly better than Gabor. Note also that the accuracy of the methods by the best k is much better than that of ($k =$ the set cardinality as in [26]). The polynomial kernel delivered the best accuracy among the three kernels for all the cases.

Table 2. Performance of individual classifiers against the sum, product, weighted sum(W-Sum), minimum score(Min), maximum score(max) least square methods. N^* is the number of images in the set.

Classifier	k = N*	k = 50	k = 25	k = 15	k = 10	k = 5
C1	89.0	90.9	91.1	91.1	90.9	90.2
C2	21.0	71.5	85.6	89.1	90.7	90.9
C3	91.2	91.8	91.6	91.2	90.8	89.9
C4	87.6	91.6	91.8	91.5	90.4	85.2
C5	80.3	91.4	92.1	92.5	92.0	87.1
C6	91.7	93.6	94.2	94.3	93.8	92.7
Min	91.2	91.8	91.6	91.2	90.8	89.9
Max	22.1	81.2	90.9	92.2	92.0	86.9
W-Sum	92.1	93.7	93.8	93.6	93.5	93.0
Product	91.1	93.6	93.8	93.8	93.5	93.2
Sum	88.7	93.3	94.2	93.8	94.1	93.6
LS	92.9	94.5	95.0	95.2	95.2	94.0

5.2 Multiple Classifier System for KPA

For the multiple classifier system the six base classifiers (2 features * 3 kernels) were considered: C1: Gaussian kernel with Gabor feature, C2: Fractional power kernel with Gabor feature, C3: Polynomial kernel with Gabor feature, C4: Gaussian kernel with LBP feature, C5: Fractional power kernel with LBP feature, C6: Polynomial kernel with LBP feature. The kernel hyper-parameters were set to perform best for the test set. The product, sum, weighted sum, minimum score and maximum score rules along with the proposed least square solution (LS) were compared in the multiple classifier systems. Table 2 shows the accuracies when the number of training images was 500. As shown our Least square formulation outperforms the individual classifiers and the baseline fusion methods for all k 's.

5.3 Random Manifold Forests

For convenience, only LBP was exploited in the experiment for Random Manifold techniques. However, by using the type of features (i.e. LBP or Gabor) as one of the random choices at a split node, better recognition accuracy may be achieved. In this experiment, we considered the following choices: a kernel type, a kernel

Table 3. Performance of RF and RtRF with LBP features for four choices of training data. ‘Raw’ represents the best performance using raw pixel images with different kernels and $k = \text{No. of images in the set (as in [26])}$, ‘Max-R’ represents the best performance using raw pixels with different kernels across k , ‘Max-LBP’ represents the best performance using LBP features with different kernels across k , ‘sum’ stands for sum rule (LBP only) and ‘LS’ is for the least square method (LBP only).

Training Data	Without Random Face Subspace(RFS)						With RFS		
	Raw	Max-R	Max-LBP	Sum	LS	RF	RtRF	RF	RtRF
1)	61.11	76.98	92.86	91.27	95.24	94.44	96.03	95.24	96.84
2)	47.20	75.20	86.40	81.60	87.20	91.20	92.00	92.80	92.80
3)	55.22	82.09	88.81	88.81	90.30	87.31	89.55	88.06	90.30
4)	59.50	90.08	94.21	93.39	95.04	92.56	94.21	92.56	94.21

hyper-parameter, a subject. For the reference set we randomly chose 500 images as explained earlier. For this experiment only 5 subjects were used from the sitcom database. Due to time and space complexity, we restricted the number of choices for kernel hyper-parameters to the following values: for Gaussian kernel: σ varies from 0.5 – 1.4 in steps of 0.1, for Fractional Power Kernel: a varies from 0.5 – 0.95 in steps of 0.05, for Polynomial Kernel: b varies from 1 – 5 in steps of 1. The total number of choices (M) is thus given by: number of kernels and its hyper-parameter choices $(10 + 10 + 5) \times$ reference sets $(5) = 125$. We also considered the random face subspace dimension and have shown the results separately for this. To set the same number of random choices for different kernels, we considered ten different random face subspace dimensions for each hyper-parameter choice from the Gaussian and Fractional kernels and twenty different random face subspace dimensions for each hyper-parameter in Polynomial kernel. In this case the total number of choices is: $(10 * 10 + 10 * 10 + 5 * 20) * 5 = 1500$.

As mentioned in Section 4.1, we considered another setting to increase the diversity among the trees and denoted the method RtRF. The following threshold values were experimented: 0.1–0.95 in steps of 0.05, i.e a total of 18 choices. Thus the total number of choices in this case is $125 * 18 = 2250$ without the random face subspace and $1500 * 18 = 27000$ with the random face subspace. Table 3 shows the performance of these two settings for different choices of training data as the reference set.

5.4 Constrained Random Manifold Forests

In the previous subsection we compared RF results using out-of-bag Error. But for a fair comparison with SVM techniques we have split the data into two sets training/testing. We have used 25 face sets per subject for training and 10 for testing. For the kernel gram matrix in SVM we used the KPA measure between two sets as given in equation 1, i.e KPA is taken as a kernel between two sets. The best kernel parameters were manually set for the SVM in the experiments. We compare this with the RtRF setup which gave the best result in Table 3. We

Table 4. Comparison on Sitcom database. CRMF-1 refers to a single reference set and CMRF-4 refers to 4 reference sets CS and CSD denote the number of constraint subspaces and the number of constraint subspace dimensions used, respectively.

Training Data	CMSM	SVM		RtRF with RFS		CRMF - 1		CRMF - 4	
		Linear	Non-L	Linear	Non-L	1CS,1CSD	5CS,5CSD	1CS,1CSD	5CS,5CSD
1)	81	95	96	89	92	97	100	99	99
				85.2	90	96.9	99.1	99	99
2)	89	91	96	92	92	93	97	98	99
				88.88	90.1	92.1	96.6	97.5	99
3)	90	93	94	87	90	90	93	96	96
				82.27	88.5	87.9	92.4	94.6	95.5
4)	86	94	96	91	93	93	95	98	98
				87.54	90.7	91.4	94.2	98	98
Avg	86.71	93.71	96.14	86.6	89.28	92.43	95.74	97.31	97.73

also compare this while using linear kernels where the random choice at every node are: 1) subject(10) 2) random Face subspace(20) 3) threshold(18). Thus the total number of choices at every node is $10 * 20 * 18 = 3600$. As mentioned in Section 4.2 we use discriminative information from CMSM. The constraint spaces are constructed using randomly selected 10 sets per subject. For space and time constraints we restrict the choices for constraint spaces to five possible subspace which are initially computed. We also restrict the dimension of these subspaces to 50% – 90% (in steps of 10%) of the total (least)significant eigenvalues. To increase flexibility we consider multiple references sets per subject each of which is obtained by randomly choosing 500 images from the training data as explained earlier. Thus the choices at every node are: 1) subject(10) 2) reference set(4) 3) constraint subspace(5) 4) constraint subspace dimension(5). Thus the total choices are $10 * 4 * 5 * 5 = 1000$. Results are shown in Table 4 and compared with SVM(1-vs-1) and original CMSM. For CMSM, from the training set we randomly choose 500 images as reference and rest are used to build the constraint space. Face subspace dimension(k) is kept as 30 and constraint subspace dimension as 90% of the (least)significant eigenvalues. 10 different trails are considered for RMF and the best along with the average performance is quoted.

For validation purpose, we also show results of the constraint random manifold forests on the Oxford dataset. Here we use raw pixel intensities, face subspace dimension(k) is kept at 10 and constraint sapce dimension is 90% of the (least)significant eigenvalues.

We have successfully included discriminative information(CMSM) in the linear version of random manifold forest and further improved its accuracy. Taking Non-Linear SVM as the baseline we get an average increase of 4% on the sitcom dataset and 32% on the Oxford database. Note that the best kernels were manually set for the SVM whereas they were automatically learnt in the proposed method.

Table 5. Comparison on Oxford database. CRMF-1 refers to a single reference set and CMRF-4 refers to 4 reference sets. CS and CSD denote the number of constraint subspaces and the number of constraint subspace dimensions used respectively.

Training Data	CMSM	SVM Non-L	CRMF - 1		CRMF - 4	
			1CS,1CSD	5CS,5CSD	1CS,1CSD	5CS,5CSD
1)	87.5	75	83.3	91.67	87.5	100
			81.67	88.75	86.67	99.58
2)	91.67	66.67	83.3	95.83	100	100
			82.08	91.67	99.17	99.58
3)	87.5	70.83	91.67	100	95.83	100
			87.92	97.08	92.92	100
4)	91.67	66.67	87.5	95.83	100	100
			84.17	93.33	96.67	100
Avg	85.12	67.26	82.74	92.92	93.04	99.7

6 Conclusions

In this paper we have explored the use of different features/kernels for KPA-based face recognition and have shown that the accuracy of the KPA method is significantly improved by using the sparse representations such as LBP and Gabor features. A novel least square formulation has been proposed for combining multiple classifiers and it has been shown to outperform some of the existing combining techniques. Both, the proposed and previous combining methods require setting the kernel hyper-parameters a priori, which is difficult in practice. To address this, we propose Random Manifold Forests that is learnt to combine discriminating regions obtained from different spaces (parameterized by the kernel type and its hyper-parameters). Hence, the method automatically learns the kernels and hyper-parameters. Taking the KPA with the raw-pixel representation as a base line, we have achieved the accuracy improvement by about *35 percent* on the challenging sitcom data set. We have successfully included discriminative information (CMSM) in the linear version of random manifold forests and further improved its accuracy. Compared to the non-linear SVM, whose kernels were manually tuned, we obtained an average increase of 4% on the sitcom dataset and 32% on the Oxford dataset.

References

1. Bosch, A., Zisserman, A., Munoz, X.: Image classification using random forests and ferns. In: ICCV (2007)
2. Breiman, L.: Random forests. Journal of Machine learning 45(1), 5–32 (2001)
3. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Journal of Machine Learning 63(1) (2006)

4. Chan, C., Kittler, J., Messer, K.: Multi-scale local binary pattern histograms for face recognition. In: ICBA (2007)
5. Chen, K., Xu, L., Chi, H.: Improved learning algorithms for mixture of experts in multiclass classification. *Neural Networks* 12(9), 1229–1252 (1999)
6. Czyz, J., Vandendorpe, L., Kittler, J.: Combining face verification experts. In: ICPR (2002)
7. Diaz-Uriarte, R., de Andrés, A.: Gene selection and classification of microarray data using random forest. *BMC bioinformatics* 7(1), 3 (2006)
8. Fukui, K., Yamaguchi, O.: Face Recognition Using Multi-viewpoint Patterns for Robot Vision. In: *Int. Sym. of Robotics Research* (2003)
9. Fukui, K., Yamaguchi, O.: The kernel orthogonal mutual subspace method and its application to 3D object recognition. In: Yagi, Y., Kang, S.B., Kweon, I.S., Zha, H. (eds.) *ACCV 2007, Part II. LNCS*, vol. 4844, pp. 467–476. Springer, Heidelberg (2007)
10. Jacobs, R., Jordan, M., Nowlan, S., Hinton, G.: Adaptive mixtures of local experts. *Neural computation* 3(1), 79–87 (1991)
11. Jain, A., Nandakumar, K., Ross, A.: Score normalization in multimodal biometric systems. *Pattern Recognition* 38(12), 2270–2285 (2005)
12. Jobson, D., Rahman, Z., Woodell, G.: A multiscale retinex for bridging the gap between color images and the human observation of scenes. *IEEE TIP* 6(7), 965–976 (1997)
13. Kapoor, A., Qi, Y., Ahn, H., Picard, R.: Hyperparameter and kernel learning for graph based semi-supervised classification. In: *NIPS* (2006)
14. Arandjelovic, O., Cipolla, R.: Automatic Cast Listing in Feature-Length Films with Anisotropic Manifold Space. In: *CVPR* (2006)
15. Kim, T.-K., Arandjelović, O., Cipolla, R.: Boosted manifold principal angles for image set-based recognition. *Pattern Recogn* 40(9), 2475–2484 (2007)
16. Kittler, J., Hatef, M., Duin, R.P., Matas, J.: On combining classifiers. *IEEE Trans. on PAMI* 20(3), 226–239 (1998)
17. Lanckriet, G., Cristianini, N., Bartlett, P., Ghaoui, L., Jordan, M.: Learning the kernel matrix with semidefinite programming. *JMLR* 5, 27–72 (2004)
18. Sonnenburg, S., Ratsch, G., Schafer, C., Scholkopf, B.: Large Scale Multiple Kernel Learning. *Journal of Machine Learning Research* 7, 1531–1565 (2006)
19. Zien, A., Ong, C.S.: Multiclass Multiple Kernel Learning. In: *ICML* (2007)
20. Lee, K., Ho, J., Kriegman, D.: Acquiring linear subspaces for face recognition under variable lighting. *IEEE Trans. on PAMI*, 684–698 (2005)
21. Rajagopal, A., Anandathirtha, P., Ramakrishnan, K., Kankanhalli, M.: Integrated Detect-Track Framework for Multi-view Face Detection in Video. In: *ICVGIP* (2008)
22. Scholkopf, B., Smola, A., Muller, K.: Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation* 10(5), 1299–1319 (1998)
23. Shotton, J., Johnson, M., Cipolla, R., Center, T., Kawasaki, J.: Semantic texton forests for image categorization and segmentation. In: *CVPR* (2008)
24. Statnikov, A., Wang, L., Aliferis, C.: A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. *BMC bioinformatics* 9(1), 319 (2008)
25. Wang, X., Tang, X.: Random sampling for subspace face recognition. *IJCV* 70(1), 91–104 (2006)
26. Wolf, L., Shashua, A.: Learning over sets using kernel principal angles. *Journal of Machine Learning Research* 4, 913–931 (2003)

27. Xiaoyu Wang, S.Y., Han, T.X.: An HOG-LBP Human Detector with Partial Occlusion Handling. In: ICCV (2009)
28. Yamaguchi, O., Fukui, K., Maeda, K.-i.: Face recognition using temporal image sequence. In: AFG (1998)
29. Nishiyama, M., Yamaguchi, O., Fukui, K.: Face recognition with the multiple constrained mutual subspace method. In: ACCV (2005)
30. Li, X., Fukui, K., Zheng, N.: Boosting Constrained Mutual Subspace Method for Robust Image-Set Based Object Recognition. IJCAI, 1132–1137 (2009)
31. Wang, R., Shan, S., Chen, X., Gao, W.: Manifold-manifold distance with application to face recognition based on image set. In: CVPR (2008)
32. Kim, T.-K., Kittler, J., Cipolla, R.: Discriminative Learning and Recognition of Image Set Classes Using Canonical Correlations. IEEE TPAMI 29(6) (2007)
33. Cevikalp, H., Triggs, B.: Face Recognition Based on Image Sets. In: Computer Vision and Pattern Recognition (2010)

Estimating Meteorological Visibility Using Cameras: A Probabilistic Model-Driven Approach

Nicolas Hautière¹, Raouf Babari¹, Éric Dumont¹,
Roland Brémond¹, and Nicolas Paparoditis²

¹ Université Paris-Est, LEPSIS, INRETS-LCPC
58 boulevard Lefebvre, F-75015 Paris

² Université Paris-Est, MATIS, IGN
73 avenue de Paris, F-94160 Saint-Mand
nicolas.hautiere@lcpc.fr

Abstract. Estimating the atmospheric or meteorological visibility distance is very important for air and ground transport safety, as well as for air quality. However, there is no holistic approach to tackle the problem by camera. Most existing methods are data-driven approaches, which perform a linear regression between the contrast in the scene and the visual range estimated by means of reference additional sensors. In this paper, we propose a probabilistic model-based approach which takes into account the distribution of contrasts in the scene. It is robust to illumination variations in the scene by taking into account the Lambertian surfaces. To evaluate our model, meteorological ground truth data were collected, showing very promising results. This work opens new perspectives in the computer vision community dealing with environmental issues.

1 Introduction

Estimating the atmospheric or meteorological visibility distance is very important for transport safety and for air quality monitoring. Dedicated optical sensors exist but are very expensive. For this reason, they are deployed only in crucial places like airports. Thus, the use of outdoor cameras is of great interest since they are low cost and already deployed for other purposes like showing current traffic and weather conditions [\[1\]](#).

Some attempts are reported in the literature to estimate the visibility using outdoor cameras or webcams. However, the visibility range differs from one application to another, so that there is no holistic approach to tackle the problem by camera. For road safety applications, the range 0-400 m is usually considered. For meteorological observation and airport safety, the range 0-1000 m is usually considered. Visual range is also used for monitoring pollution in urban areas. In this case, higher visual ranges, typically 1-5 km, are usually considered. In this paper, we propose a method which copes with the different application constraints.

Two families of methods are proposed in the literature. The first one estimates the maximum distance at which a selected target can be seen. The methods differ depending on the nature of the target and how to estimate the distance. For intelligent vehicles as well as for visual monitoring of highway traffic, a black target at the horizon is chosen and a flat road is assumed. Bush [2] uses a wavelet transform to detect the highest edge in the image with a contrast above 5%. Based on a highway meteorology standard, Hautière et al. [3] proposed a reference-free roadside camera-based sensor which not only estimates the visibility range but also detects that the visibility reduction is caused by fog. For meteorological observations, regions of interest whose distance can be obtained on standard geographic maps are selected manually [4]. An accurate geometric calibration of the camera is necessary to operate these methods.

A second family of methods correlates the contrast in the scene with the visual range estimated by reference additional sensors [5]. No accurate geometric calibration is necessary. Conversely, a learning phase is needed to estimate the function which maps the contrast in the scene to the visual range. The method proposed in this paper belongs to this second family. Usually, a simple gradient based on the Sobel filter or a high-pass filter in the frequency domain are used to compute the contrast [6–8]. Luo et al. [9] have shown that the visual range obtained with both approaches are highly correlated. Liaw et al. [6] proposed to use a homomorphic filter in addition to the high-pass filter in order to reduce the effects of non-uniform illumination. Once the contrast is computed, a linear regression is performed to estimate the mapping function [5, 6, 8]. Due to this step of linear regression, these methods can be seen as data driven approaches.

Unlike previous data-driven approaches, we propose a probabilistic model-driven approach which allows computing a physics-based mapping function. Unlike existing approaches, our model is non-linear, which allows encompassing the whole spectrum of applications. In particular, our model takes into account the distribution of contrasts in the scene. Unlike existing approaches, e.g. [8], our model is robust to illumination variations in the scene by taking into account the physical properties of objects in the scene. To assess the relevance of our approach, we have collected ground truth data. Using these rare experimental data, we are able to present very promising results, which might open new trends in the computer vision community dealing with environmental issues.

The remainder of this paper is organized as following. In section 2, we recall the Koschmieder’s model of fog visual effects on which we base our work. In section 3, we present a model-driven approach, whose experimental evaluation is carried out in section 4. Finally, we discuss the results and conclude.

2 Vision through the Atmosphere

The attenuation of luminance through the atmosphere was studied by Koschmieder [10], who derived an equation relating the extinction coefficient of the atmosphere β , the apparent luminance L of an object located at distance d , and the luminance L_0 measured close to this object:

$$L = L_0e^{-\beta d} + L_\infty(1 - e^{-\beta d}) \tag{1}$$

(1) indicates that the luminance of the object seen through fog is attenuated by $e^{-\beta d}$ (Beer-Lambert law); it also reveals a luminance reinforcement of the form $L_\infty(1 - e^{-\beta d})$ resulting from daylight scattered by the slab of fog between the object and the observer, the so-called airlight. L_∞ is the atmospheric luminance.

On the basis of this equation, Duntley developed a contrast attenuation law (10), stating that a nearby object exhibiting contrast C_0 with the fog in the background will be perceived at distance d with the following contrast:

$$C = \left[\frac{L_0 - L_\infty}{L_\infty} \right] e^{-\beta d} = C_0e^{-\beta d} \tag{2}$$

This expression serves to base the definition of a standard dimension called meteorological visibility distance V , i.e. the greatest distance at which a black object ($C_0 = -1$) of a suitable dimension can be seen on the horizon, with the threshold contrast set at 5% (11). It is thus a standard parameter that characterizes the opacity of a fog layer. This definition yields the following expression:

$$V \approx \frac{3}{\beta} \tag{3}$$

More recently, Koschmieder’s model has received a lot of attention in the computer vision community, e.g. (12-17). Indeed, thanks to this model, it is possible to infer the 3D structure of a scene in fog presence, or to dehaze/defog images by reversing the model. However, it is worth mentioning that in these works a relative estimation of the meteorological visibility is enough to restore the visibility. In this paper, we use Koschmieder’s model to estimate the actual meteorological visibility distance, which makes the problem quite different.

3 The Model-Driven Approach

3.1 Contrast of a Distant Target

Assuming a linear response function of the camera, the intensity I of a distant point located at distance d in an outdoor scene is given by Koschmieder’s model (1):

$$I = Re^{-\beta d} + A_\infty(1 - e^{-\beta d}) \tag{4}$$

where R is the intrinsic intensity of the pixel, i.e. the intensity corresponding to the intrinsic luminance value of the corresponding scene point and A_∞ is the background sky intensity. Two points located at roughly the same distance $d_1 \approx d_2 = d$ with different intensities $I_1 \neq I_2$ form a distant target whose normalized contrast is given by:

$$C = \frac{I_2 - I_1}{A_\infty} = \left[\frac{R_2 - R_1}{A_\infty} \right] e^{-\beta d} = C_0e^{-\beta d} \tag{5}$$

In this equation, the contrast C of a target located at d depends on $V = \frac{3}{\beta}$ and on its intrinsic contrast C_0 . If we now assume that the surface of the target is Lambertian, the luminance L at each point i of the target is given by:

$$L = \rho_i \frac{E}{\pi} \tag{6}$$

where ρ_i denotes the albedo at i . Moreover, it is a classical assumption to set $L_\infty = \frac{E}{\pi}$ so that (5) finally becomes:

$$C = (\rho_2 - \rho_1)e^{-\beta d} \approx (\rho_2 - \rho_1)e^{-\frac{3d}{V}} = \Delta\rho \times e^{-\frac{3d}{V}} \tag{7}$$

Consequently, the contrast of a distant Lambertian target only depends on its physical properties and on its distance to the sensor and on the meteorological visibility distance, and no longer on the illumination. These surfaces are robust to strong illumination variations in the computation of the contrast in the scene.

3.2 Probabilistic Modeling

Let us consider an outdoor scene where targets are distributed at different distances from the camera. Let us denote ϕ the probability density function of observing a contrast C in the scene:

$$\mathbb{P}(C < X \leq C + \text{crf}C) = \phi(C)\text{crf}C \tag{8}$$

We denote ψ the p.d.f. of there being a target at the distance d . Thanks to (7), ϕ can be written as a function of ψ :

$$\phi(C) = -\frac{V}{3C\Delta\rho}\psi(d) \tag{9}$$

The mean contrast in the scene can thus be computed thanks to the density of targets in the scene:

$$m = \int_0^1 C\phi(C)\text{crf}C = \int_0^{+\infty} \Delta\rho\psi(d)e^{-\frac{3d}{V}}\text{crf}d \tag{10}$$

To express m , a realistic expression for the density of targets ψ in the scene is needed.

3.3 Expectation of the Mean Contrast

In this paragraph, we search an analytical expression of (10). In this aim, we assume a scene which contains n Lambertian targets with random albedos located at random distances between 0 and d_{\max} . For a given sample scene, we can compute the mean contrast of the targets with respect to the meteorological visibility distance and plot the corresponding curve. Some sample curves are plotted in

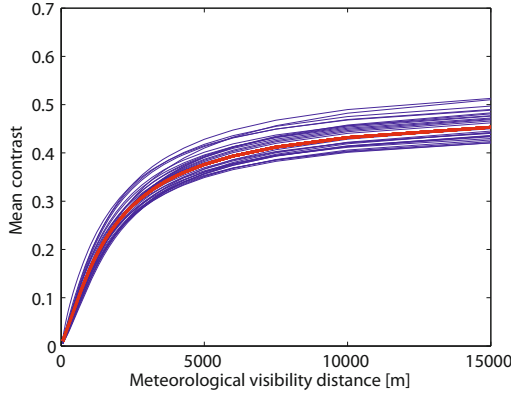


Fig. 1. Blue: curves depicting the mean contrast in random scenes with respect to the meteorological visibility distance. Red: expectation of the mean contrast.

blue in Fig. 1 ($n = 100$ and $d_{\max} = 1000$ m). We can compute the mathematical expectation of the mean contrast and obtain the following analytical model:

$$m_u = \frac{V \Delta \bar{\rho}}{6d_{\max}} \left[1 - \exp \left(- \frac{3d_{\max}}{V} \right) \right] \quad (11)$$

where $\Delta \bar{\rho}$ is the mean albedo difference of the targets in the scene. We plot this model in red in Fig. 1. If we do not have any a priori on the targets distribution in the scene, this analytical model is the most probable with which to fit the data. That will be experimentally assessed in section 4.

At this stage, we can make a comparison with the charging/discharging of a capacitor. The capacitance of the system is determined by the distribution of Lambertian targets in the scene. The smaller the capacitance of the system is, the faster the curves go to 0.5. We thus define an indicator τ of the system quality which is the meteorological visibility distance at which two thirds of the "capacitance" is reached. A high value of τ also means a lower sensitivity of the model at low meteorological visibility distances.

3.4 Model Inversion and Error Estimation

In the previous section, we have computed an analytical expression of the mean contrast expectation m_u with respect to the meteorological visibility distance V . Ultimately, we would like to compute V as a function of m_u . In this aim, we need to invert the mean contrast expectation function (11). The inversion of this model exists and is expressed by:

$$V(m_u) = \frac{3m_u d_{\max}}{1 + m_u W \left(\frac{e^{-\frac{1}{m_u}}}{m_u} \right)} \quad (12)$$

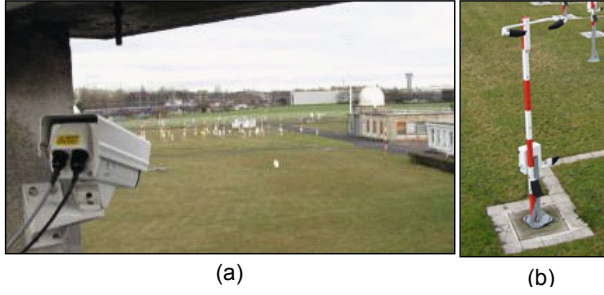


Fig. 2. Instrumentation of our observation field test: (a) the camera grabbing pictures of the field test; (b) the scatterometer along with the background luminancemeter

where Lambert W is a transcendental function defined by solutions of the equation $We^W = x$ [18].

4 Experimental Evaluation

In this section, we present an experimental evaluation of the proposed model for visibility estimation. In this aim, we have collected ground truth data. First, we present the methodology. Second, we present our method to estimate whether a surface is Lambertian or not. Third, we present the results.

4.1 Methodology

Instrumentation. The observation field test we used is equipped with a reference transmissometer (Degreane Horizon TI8510). It serves to calibrate different scatterometers (Degreane Horizon DF320) used to monitor the meteorological visibility distance on the French territory, one of which provided our data. They are coupled with a background luminance sensor (Degreane Horizon LU320) which monitors the illumination received by the sensor. We have added a camera which grabs images of the field test every ten minutes. The camera is an 8-bit CCD camera (640×480 definition, $H=8.3$ m, $\theta = 9.8^\circ$, $f_l = 4$ mm and $t_{pix} = 9 \mu m$). It is thus a low cost camera which is representative of common video surveillance cameras. Fig. 2(a) shows the installed camera and its orientation with respect to the field test. Fig. 2(b) shows the scatterometer and the background luminancemeter.

Data Collection. We have collected two fog events at the end of February 2009. The fog occurred early in the morning and lasted a few hours after sunrise. During the same days, there were strong sunny weather periods. Fig. 3 shows sample images. Figs. 3(a) is a sample of sunny weather. Fig. 3(b) is a sample of cloudy weather. Fig. 3(c) is a sample of foggy weather. The corresponding

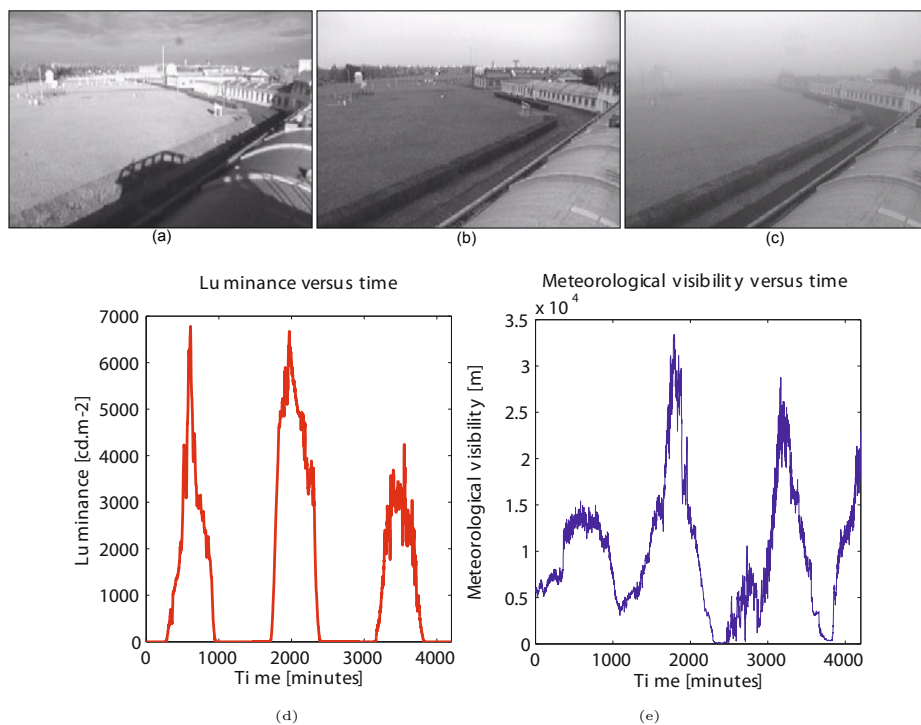


Fig. 3. Samples of data collected in winter 2008-2009: (a) images with strong illumination conditions and presence of shadows; (b) cloudy conditions; (c) foggy weather situation; (d) meteorological visibility distance data and (e) background luminance data collected in the field test during two days

meteorological visibility distances and luminances are plotted in Fig. 3(d,e). As one can see, the meteorological visibility distance ranges from 100 m to 35.000 m and the luminance ranges from 0 to 6.000 cd.m^{-2} .

We have thus collected quite rare experimental data. Indeed, during a short period of time, we had rapidly changing weather conditions. The ranges of meteorological visibility distance and luminance were very large. In the literature, works are dedicated to limited ranges of visibility distances. For example, road safety applications are dealing with 0-400 m whereas people working on environmental issues are dealing with meteorological visibility distances which are above 1000 m. We are among the first to have collected data encompassing both ranges. Moreover, since the data were collected in a short period of time, we can consider that the content of the scene did not change. For example, we can consider that the phenology of the trees did not change, so that the amount of texture in the scene without fog remains constant. This database is available on LCPC's web site <http://www.lcpc.fr/en/produits/matilda/> for research purpose.

4.2 Location of Lambertian Surfaces

To estimate m and thus V , we compute the normalized gradient only on the Lambertian surfaces of the scene as proposed in section 3.1. We thus need to locate Lambertian surfaces in the images. In this aim, we compute the Pearson coefficient, denoted $P_{i,j}^L$, between the intensity of pixels in image series where the position of the sun changes and the value of the background luminance estimated by the luminancemeter. The closer $P_{i,j}^L$ is to 1, the stronger the probability that the pixel belongs to a Lambertian surface. This technique provides an efficient way to locate the Lambertian surfaces in the scene. For our field test, the mask of Lambertian surfaces is shown in Fig. 4. The redder the pixel, the more the surface is assumed to be Lambertian.

Having located the Lambertian surfaces, we can compute the gradients in the scene by means of the module of the Sobel filter. For each pixel, we normalize the gradient $G_{i,j}$ by the intensity of the background. Since our camera is equipped with an auto-iris, the background intensity A_∞ is most of the time equal to $2^8 - 1$, so that this step can be avoided. Each gradient is then weighted by $P_{i,j}^L$, the probability of a pixel to belong to a Lambertian surface where no depth discontinuity exists (P^L is mostly very small). Consequently, only relevant areas of the image are used, and the scene need not be totally Lambertian. Finally, the estimated contrast in the scene \tilde{m}_u is given by:

$$\tilde{m}_u = \sum_{i=0}^H \sum_{j=0}^W \Delta\rho_{i,j} \exp\left(-\frac{3d_{i,j}}{V}\right) \approx \sum_{i=0}^H \sum_{j=0}^W \frac{G_{i,j}}{A_\infty} P_{i,j}^L \quad (13)$$

where $\Delta\rho_{i,j}$ is the intrinsic contrast of a pixel (7), and H and W are respectively the height and the width of the images. Finally, the approach makes the best use of the physics of the scene and would be able to process scenes without any Lambertian surfaces (at the cost of lowering the quality of the results).



Fig. 4. Mask of Lambertian surfaces on our field test: The redder the pixel is, the higher the confidence that the surface is Lambertian

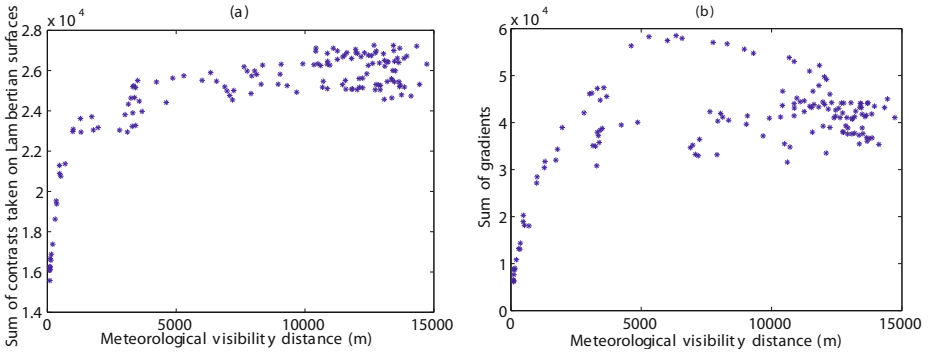


Fig. 5. Visibility estimators: (a) the estimator is based on the contrast on Lambertian surfaces; (b) the estimator is only based on Sobel’s gradient module

4.3 Results

Contrast Estimators. We have computed (13) for our collection of 150 images with different meteorological visibility distances. For comparison purposes, we have also computed the simple sum of gradients in the image without taking into account the segmentation of the Lambertian surfaces. The results are shown in Fig. 5. Using the Lambertian surfaces, we can see that the shape of the distribution in Fig. 5(a) looks like the curve proposed in Fig. 1, which is very satisfactory. Conversely, when all the pixels of the scene are used, the points are more scattered when the meteorological visibility distance is above 2500 m (see Fig. 5(b)). When the visibility is above 2500 m, the illumination by the sun does influence a lot the gradients in the scene. When the weather is sunny, i.e. the visibility is better, the influence of the sun is more important so that the gradient is changing with respect to the sun position. Consequently, the estimation of the visibility is altered. These two distributions show the benefit of selecting the Lambertian surfaces to estimate the visibility distance.

Model Fitting. We have to fit the mean contrast model (11) to the data shown in Fig. 5(a) using robust regression techniques. To ensure a mathematical solution, we have fitted the model (14), which is slightly different from the theoretical model. Three unknown variables a , b and d_{max} have to be estimated, which can be easily done using classical curve fitting tools.

$$\tilde{m}_u = \frac{aV}{d_{max}} \left[1 - \exp \left(- \frac{3d_{max}}{V} \right) \right] + b \tag{14}$$

This model fits well with the data ($R^2 = 0.91$). In particular, we obtain $d_{max} = 307.2$ m. The fitted curve is plotted in Fig. 6. We estimated a capacitance of the system $\tau \approx 950$ m.

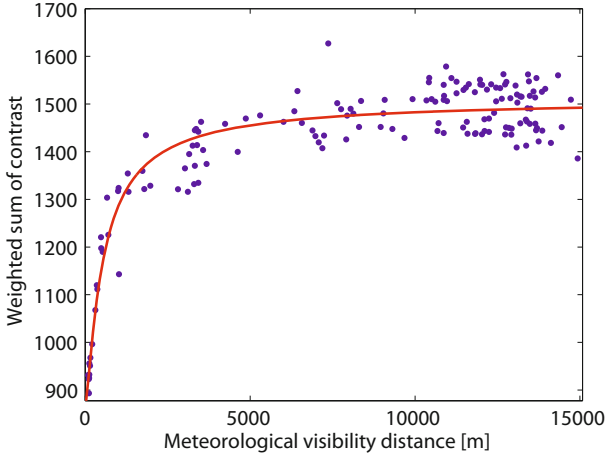


Fig. 6. Data fitting with the mean contrast model. Dots: data. Red curve: fitted model.

Discussions. From the fitted model, we can now invert the model using (12) and estimate the meteorological visibility distance \tilde{V} based on the mean contrast m_u :

$$\tilde{V} = \frac{3d_{\max}(b - m_u)}{(b - m_u)W\left(\frac{ae^{\frac{a}{b-m_u}}}{b - m_u}\right) - a} \tag{15}$$

Having estimated the meteorological visibility distance, we can compute the error on this estimation. The results are given in Table 1. Since the applications are very different depending on the range of meteorological visibility distances, we have computed the error and the standard deviation for various applications: road safety, meteorological observation and air quality. One can see that the error remains low for critical safety applications. It increases for higher ranges of visibility distances, and becomes huge for visibility distances above 7 km.

Table 1. Relative errors of meteorological visibility distance estimation with respect to the envisaged application

Application	Highway fog	Meteorological fog	Haze	Air quality
Range [m]	0-400	0-1000	0-5000	0-15000
Number of data	13	19	45	150
Mean error [%]	12.6	18.1	29.7	-
Std [%]	13.7	18.9	22	-

Different points may be discussed. First, the model used in this paper is relevant for uniform distribution of distances which happen in many environments, such as highway scenes. The scene from which the experimental data used in this

paper are issued may be not meet this assumption. Second, the Sobel operator is certainly not the best estimate for the gradient. Indeed, it is a simple high-pass filter which is problematic because of the impulse noise of camera sensors. Different filters may be used to beforehand enhance the images, or to compute the contrast more robustly.

5 Conclusion

In this paper, we propose a probabilistic model-driven approach to estimate the meteorological visibility distance through use of generic outdoor cameras based on a mean contrast expectation function. Unlike previous data-driven approaches, we use a physical model which allows computing a mapping function between the contrast and the meteorological visibility estimated by an additional reference sensor. Our model is non-linear which allows dealing with a large spectrum of applications. The calibration of our system is less sensitive to the input data due to its intrinsic physical constraints. In particular, our model takes into account the distribution of targets in the scene. It is also robust to illumination variations in the scene by taking into account the Lambertian surfaces. To evaluate the relevance of our approach, we have collected ground truth data with the help of our national meteorological institute. Using these rare experimental data, we obtain promising results. In future work, we intend to estimate the contrast expectation function without any additional meteorological sensor, based only on the properties of the scene (geometry, texture) collected by remote sensing techniques and the characteristics of the camera. Such a model-driven approach paves the road to methods without any learning phases.

References

1. Jacobs, N., Burgin, W., Fridrich, N., Abrams, A., Miskell, K., Braswell, B., Richardson, A., Pless, R.: The global network of outdoor webcams: Properties and applications. In: ACM International Conference on Advances in Geographic Information Systems, ACM GIS 2009 (2009)
2. Bush, C., Debes, E.: Wavelet transform for analyzing fog visibility. *IEEE Intelligent Systems* 13(6), 66–71 (1998)
3. Hautière, N., Bigorgne, E., Bossu, J., Aubert, D.: Meteorological conditions processing for vision-based traffic monitoring. In: International Workshop on Visual Surveillance, European Conference on Computer Vision (2008)
4. Bäumer, D., Versick, S., Vogel, B.: Determination of the visibility using a digital panorama camera. *Atmospheric Environment* 42, 2593–2602 (2008)
5. Hallowell, R., Matthews, M., Pisano, P.: An automated visibility detection algorithm utilizing camera imagery. In: 23rd Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology (IIPS), San Antonio, TX, Amer. Meteor. Soc (2007)
6. Liaw, J.J., Lian, S.B., Chen, R.C.: Atmospheric visibility monitoring using digital image analysis techniques. In: Jiang, X., Petkov, N. (eds.) CAIP 2009. LNCS, vol. 5702, pp. 1204–1211. Springer, Heidelberg (2009)

7. Hagiwara, T., Ota, Y., Kaneda, Y., Nagata, Y., Araki, K.: A method of processing CCTV digital images for poor visibility identification. *Transportation Research Records* 1973, 95–104 (2007)
8. Xie, L., Chiu, A., Newsam, S.: Estimating atmospheric visibility using general-purpose cameras. In: Bebis, G. (ed.) *ISVC 2008, Part II. LNCS*, vol. 5359, pp. 356–367. Springer, Heidelberg (2008)
9. Luo, C.H., Wen, C.Y., Yuan, C.S., Liaw, J.-L., Lo, C.C., Chiu, S.H.: Investigation of urban atmospheric visibility by high-frequency extraction: Model development and field test. *Atmospheric Environment* 39, 2545–2552 (2005)
10. Middleton, W.: *Vision through the atmosphere*. University of Toronto Press (1952)
11. CIE: *International Lighting Vocabulary*. Number 17.4 (1987)
12. Narasimhan, S.G., Nayar, S.K.: Vision and the atmosphere. *Int. J. Comput. Vis.* 48(3), 233–254 (2002)
13. Narasimhan, S.G., Nayar, S.K.: Contrast restoration of weather degraded images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25(6), 713–724 (2003)
14. Hautière, N., Tarel, J.P., Aubert, D.: Towards fog-free in-vehicle vision systems through contrast restoration. In: *IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, Minnesota, USA (2007)
15. Tan, R.T.: Visibility in bad weather from a single image. In: *IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, Alaska, USA (2008)
16. He, K., Sun, J., Tang, X.: Single image haze removal using dark channel prior. In: *IEEE Conference on Computer Vision and Pattern Recognition*, Miami, Florida, USA (2009)
17. Tarel, J.P., Hautière, N.: Fast visibility restoration from a single color or gray level image. In: *IEEE International Conference on Computer Vision*, Kyoto, Japan (2009)
18. Corless, R.M., Gonnet, G.H., Hare, D.E.G., Jeffrey, D.J., Knuth, D.E.: On the Lambert W function. *Advances in Computational Mathematics* 5, 329–359 (1996)

Optimizing Visual Vocabularies Using Soft Assignment Entropies

Yubin Kuang, Kalle Åström, Lars Kopp,
Magnus Oskarsson, and Martin Byröd

Centre for Mathematical Sciences
Lund University

Abstract. The state of the art for large database object retrieval in images is based on quantizing descriptors of interest points into visual words. High similarity between matching image representations (as bags of words) is based upon the assumption that matched points in the two images end up in similar words in hard assignment or in similar representations in soft assignment techniques. In this paper we study how ground truth correspondences can be used to generate better visual vocabularies. Matching of image patches can be done e.g. using deformable models or from estimating 3D geometry. For optimization of the vocabulary, we propose minimizing the entropies of soft assignment of points. We base our clustering on hierarchical k-splits. The results from our entropy based clustering are compared with hierarchical k-means. The vocabularies have been tested on real data with decreased entropy and increased true positive rate, as well as better retrieval performance.

1 Introduction

One of the general problems in computer vision is to automate the recognition process using computer algorithms. For problems such as object recognition and image retrieval from large databases, the state of the art is based on the bags of words (BOW) framework [18, 20, 21, 23]. Firstly a set of interest points are extracted in each of the images using interest point detectors [6, 11, 13, 14] or dense sampling. Then feature descriptors e.g. SIFT or SURF [2, 11, 15] are computed at each interest point. To enable fast matching, feature descriptors are quantized into visual words as a vocabulary, where the descriptors assigned with the same word are regarded as matched. Finally, the co-occurrence of visual words between a query image and those in the database is then used to generate hypotheses of matched images. The matching is often based on the histograms of visual words and the L_1 norm or L_2 norm of differences between two histograms (or the intersection of two histograms) after normalization.

A good vocabulary in the quantization step of the BOW pipeline is crucial for the recognition and retrieval system. Traditional approaches [9, 18, 21, 23] construct vocabulary by clustering descriptor vectors derived from training images in an unsupervised way, i.e. without ground truth information on which correspondence class a specific feature belongs to. These approaches either suffer from

quantization errors or have difficulties in matching wide variety of appearances of objects in images, due to large differences in view points, lighting conditions and background clutter as well as the large intra-class variations of the objects themselves. One way to resolve this is through learning, with the presence of large amount of correspondence ground truth data. While obtaining ground truth data from raw images can be expensive, incorporating such information with proper schemes can enable efficient and accurate recognition performance.

Efforts have been made on learning vocabulary with ground truth information. Winn et al. [26] quantized features with k-means after which the resulting words were merged to obtain intra-class compactness and inter-class discrimination. On the other hand, Moosman et al. used random forests as the quantizer such that at each split an entropy measure based on the class labels is maximized [17]. In [19] Perronnin et al. used class-level labels and proposed to train class-specific vocabularies modeled by GMMs and combine them with a universal vocabulary. The most related work to ours in technical aspects, is the work by Lazebnik et al. in [10], where they simultaneously optimize the quantizer in Euclidean feature space and the posterior class distribution. All these previous works imposed the supervision such that each word in the vocabulary has a discriminative representation of the different object classes. However, they have mainly focused on object categorization and the number of class labels is relatively small (≈ 20) except for the the work in [7] introduced hidden Markov random fields for semantic embedding of local patch features with relatively large number of class labels (≈ 3600). Our approach is designed for image retrieval and uses very large scale ($\approx 80K - 250K$) partially labeled patch correspondences to quantize feature space in a hierarchical manner.

For object recognition, the learned vocabulary has to be more specific regarding matching features. Each word in the vocabulary should contain only small number of features such that each word might encode the appearance variations of a single physical point. In [16], Mikulik et al. start with an unsupervised vocabulary and apply a supervised soft-assignment afterwards, where words are connected based on the statistics of matched feature points from a huge dataset with ground truth correspondences. Another line of work [22, 24], is to incorporate the supervision into the feature metric learning before quantization such that the matched pairs of features have small distances than non-matched pairs in the new mapping. Both methods achieves substantial improvement in the retrieval tasks. Our approach works on the original feature space and encodes the ground truth correspondences in the process of vocabulary generation.

In this paper, we focus on vocabulary for recognition and would like to address systematically the following questions: (i) How large should the vocabulary be? In the current literature the sizes range from less than a thousand to millions of words in the vocabulary. This could of course be highly application dependent. (ii) How can we evaluate the quality of the vocabulary? (iii) What is the optimal division of the feature space into words and How do we avoid splitting matching features into different words? To address the first two questions, we first studied statistically how true positive rates and false positive rates in matching features

of a vocabulary affect the retrieval performance. We then present a framework for supervised vocabulary training using partial or full ground truth information on correspondences. In such a way, we obtain a vocabulary that encodes the intra-class variation of each correspondence class leading to improved retrieval performance.

The rest of the paper is organized as follows. Section 2 contains brief discussion on methods for obtaining the ground truth correspondences. In Section 3 we present the modelling of mAP from vocabulary matching statistics. In Section 4 we describe our optimization method for training the vocabulary using ground truth data. The method is then tested on real image data in Section 5.

2 Ground Truth Correspondence Data

In order to obtain a good visual vocabulary for object recognition in images, we propose to learn the vocabulary using ground truth information on corresponding points. The motivation is that we believe that this strengthens the vocabulary as opposed to just doing unsupervised clustering and we expect the gain to be worthwhile since the the more expensive training with ground truth is a off-line process in the retrieval pipeline.

In order for the learned visual vocabulary to be robust a wide variety of appearances of objects in images, the ground truth datasets should preferably present for the same physical point or same object (i) Large intra-class variability of the objects themselves. (ii) Large differences in lighting conditions. (iii) Large differences in view points. We will discuss in the following some of the methods for obtaining such data sets.

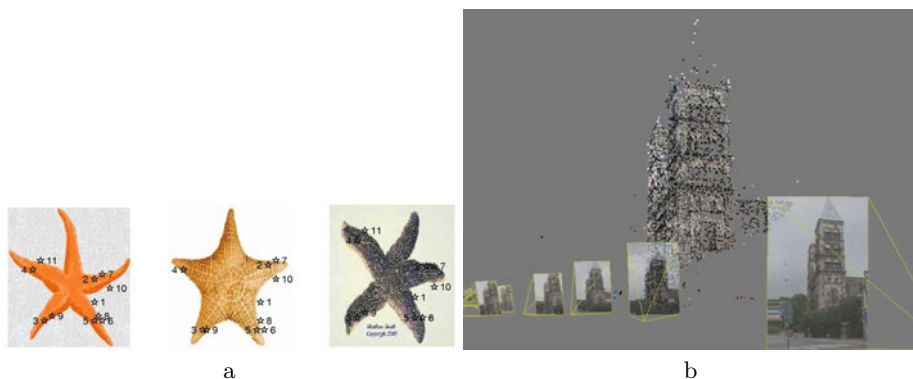


Fig. 1. Two methods of obtaining ground truth correspondences for vocabulary training using (a) deformable shape model and (b) structure and motion algorithms

For object categorization, intra-class variability that is present for most object categories that are interesting for recognition. Deformable models can be used here to generate correspondences. Training these models can be cumbersome, but we believe that this will benefit the training process of the visual vocabulary

enabling a very fast but accurate bottom up search process, that is based on learned high level features. In Figure 1(a) a deformable shape model estimated from image data is shown, where the correspondences are based on optimizing the minimum description length according to [8]. The images are from the starfish category of the Caltech-256 object category dataset [4].

The lighting variabilities could be achieved by having images of static scenes taken under substantially changing lighting conditions. View points variabilities could be obtained by estimating the geometry of objects from images taken at different view points using a RANSAC framework in combination with epipolar geometry estimation such as in e.g. [1, 5]. In Figure 1(b) the typical result from the geometry estimation is shown. From the corresponding points in the images, feature descriptors can then be extracted from the images. In [16], Mikulik et al. present an efficient way of generating large scale ground truth dataset from collections of images by image matching graph. An alternative in [24], Strecha et al. also utilize geo-tags in their 3D-reconstruction pipeline to obtain geometrically consistent patches. In the experimental section of this paper the visual vocabularies are trained on partial ground truth data obtained from the UBC Patch Data [25].

3 Modelling Mean Average Precision from Vocabulary Statistics

One key argument made in this paper is that good retrieval systems, e.g. as measured by mean Average Precision (mAP) can be obtained by studying the properties of the vocabulary on the statistics of descriptor distribution both for random (not necessarily matching) descriptor pairs and for matching descriptor pairs. By matching descriptor pairs we do not mean descriptors that end up in the same word in the vocabulary, but rather descriptors of matching interest regions, i.e. regions which are matching in a ground truth sense.

We can evaluate a vocabulary with two simple characteristics, (i) the false positive rate p_{fp} , which is the probability that two random descriptors end up in the same word and (ii) the true positive rate p_{tp} , which is the probability that two matching descriptors end up in the same word.

We argue that the mapping from true positive and false positive rates to mean average precision can be modelled and analyzed. High mean average precision is obtained using vocabularies with low false positive rates and high true positive rates.

The mapping depends on many characteristics of the test, such as the number of features in each image, the number of images in the database, the proportion of positive vs negative answers to a image retrieval query etc. In this model we have for simplicity assumed that histograms are measured with the normed L_1 distance, but other distance metrics could be used. In fact the modelling could come to good use in determining which metrics to use.

Modelling the L_1 -distance Distribution for Two Random Images

Assuming that the distribution of features in different visual words is known, and assuming that features in two random images are independent, it is possible to simulate and model the distribution of L_1 distances. In Figure 2a three such distributions are shown for small, medium and large vocabularies.

For large vocabularies the histograms are sparse. A reasonable approximation here is that the distance is $d = (2n - 2o)/n$, where n is the number of features in the images and o is the number of common features. The number of overlapping features o can be approximated reasonably using binomial distributions using n samples with probability $p = n/w$. For increasing vocabulary size this distribution is pushed towards the right end of the spectrum.

Modelling the L_1 -distance Distribution for Two Matching Images

For two matching images we assume that there are a number of matching features. For each matching feature pair there is a certain probability p_t that they end up in the same word. For the remaining features we assume that they end up in random words according to the distribution above. The resulting distribution of L_1 -distance is similar to that of two random features, but pushed slightly to the left. In Figure 2a three such distributions are shown, again for small, medium and large vocabularies.

Modelling Precision, Recall and Mean Average Precision

For each vocabulary as characterized by its true and false positive rates (p_{tp}, p_{fp}), we can estimate the probability distribution of matched image L_1 -distance, p_m , and the probability distribution of two random image L_1 -distance, p_r .

Assuming that in a random query there are N_{inlier} matching images and $N_{outlier}$ non-matching images. For each threshold D of L_1 -distances we obtain a query result with precision

$$R = \frac{N_{inlier} \int_0^D p_m(x) dx}{N_{inlier} \int_0^2 p_m(x) dx} = \int_0^D p_m(x) dx$$

and recall

$$P = \frac{N_{inlier} \int_0^D p_m(x) dx}{N_{inlier} \int_0^D p_m(x) dx + N_{outlier} \int_0^D p_r(x) dx} = \frac{\int_0^D p_m(x) dx}{\int_0^D p_m(x) dx + K \int_0^D p_r(x) dx}$$

where $K = \frac{N_{outlier}}{N_{inlier}}$ is the ratio of outliers to inliers in a typical query.

Note that the domain of the normalized L_1 distance is between $[0,2]$. Therefore, in the equation for recall, we have used 2 as the integration limit in denominator. It follows that the integral in the denominator is 1. From these two curves it is straightforward to estimate the mean average precision.

Figure 2b shows how the mean average precision depends on the 10-log of the true and false positive rates (p_{tp}, p_{fp}). Notice that this confirms the theory that quite large vocabularies are needed for good performance.

A key argument made here is that e.g. for hierarchical vocabulary building, increased levels of splitting of the vocabulary gives lower true and false positive

rates. But already for small vocabularies, by demonstrating that one obtains higher true positive rates, while retaining a low false positive rate will be beneficial for the end performance as measured by the mean average precision.

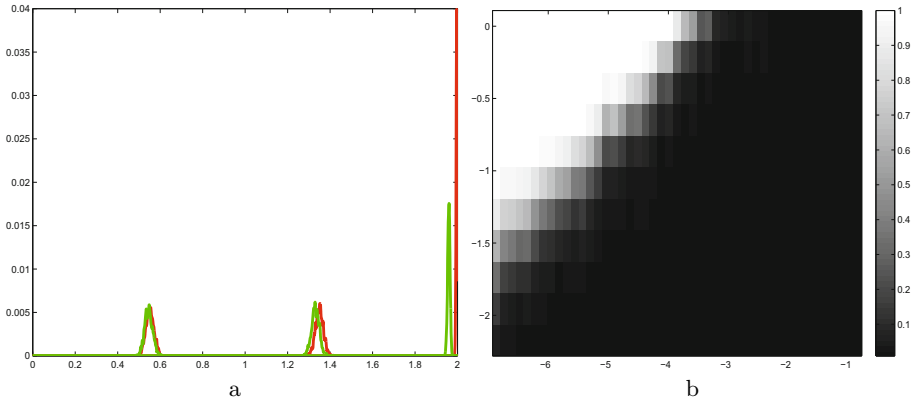


Fig. 2. (a) L_1 -distance distributions for random image pairs (red) and matching image pairs (green) for three vocabularies of different size. (b) Mean Average Precision as a function of 10-log of false positive rate (x-axis) and 10-log of true positive rate (y-axis).

4 Optimizing the Vocabulary with Respect to Entropy

We will concentrate on hierarchical divisions of the descriptor space. The resulting vocabularies have the advantage that visual word generation is extremely efficient. Another advantage during training is that the learning and corresponding optimizations only have to be done at each hierarchical split in the tree.

We assume that a number of descriptors are given, $x_i \in \mathbb{R}^d, i = 1 \dots N$, and that correspondences among such descriptors are known. Here we have represented such correspondences as the correspondence class C_i for each point i . The number of correspondence classes is denoted by N_c . In the typical datasets that we have worked on, the numbers of descriptors are in the order of 500K and the numbers of correspondence classes are in the order of 150K. The correspondences have been generated by sampling from 3D reconstructions of scenes. See Section 5.1 for details. Other ways of generating correspondences could be through geometric matching in image pairs, tracking of points in image sequences or by hand annotated data.

We will concentrate on the problem of recognizing specific scenes and the data that we have used is chosen so that points are in correspondence if they denote the same physical point in the scene. That descriptors are in different correspondence classes does not necessarily mean that they are *not* in correspondence. On the contrary, we expect there to be many descriptors in different correspondence classes that actually correspond quite well. However for points that are in correspondence we would like the corresponding descriptors to end up in the same word in the final vocabulary.

Our hierarchical division will be based on k splits in the descriptor space D at each step. Each such split is represented by k center points c_1, \dots, c_k and a scalar m that can be interpreted as a margin. Low values of m represent sharper cuts and high values represent softer classification.

We study both hard assignment and soft assignment in the following sense. For hard assignment a descriptor is put in the bin i corresponding to the closest center c_i . For soft assignment we put each point x in the k bins in proportion to the weight w_i according to

$$w_i = \frac{\exp\left(\frac{|x-c_i|}{m}\right)}{\sum_{j=1}^k \exp\left(\frac{|x-c_j|}{m}\right)}. \quad (1)$$

Contrary to [10] we use exponential distributions, which give smoothing that only depends on the difference of distances to the cluster center. Descriptors for which the distance to the closest centers is similar fall into several bins to a fair degree, whereas descriptors for distance difference between the two closest centers is much larger than m fall essentially only into one part of the tree.

4.1 Entropy Model

To optimize the division parameters $z = (c_1, \dots, c_k)$ for hard assignment and $z = (c_1, \dots, c_k, m)$ for soft assignment, we use entropy as a criterion. Entropy takes into account both that the split is balanced, i.e. that approximately equal number of descriptors fall into each bin, and that the correspondence classes are split as cleanly as possible. The entropy for a random variable X with N possible states is defined as $E = -\sum_{i=1}^N p(i) \log_2(p(i))$, where p is the probability density function of X . Here we use the 2-log as it is more intuitive and easier to interpret.

Entropy is fairly easy to use in the sense that it is straightforward to define for both hard and soft assignment. The probability density function is calculated in the following manner. In each split we calculate the (weighted) histogram of descriptors in each correspondence class $h_{tot} = (h(1), \dots, h(N_c))$ before the split. Each descriptor falls partly in the k different parts of the tree, thus contributing in part to both the k -weighted histogram h_1, \dots, h_k .

By normalizing the histograms with the sum, we obtain correspondence class probabilities, i.e. $p_{tot}(i) = \frac{h_{tot}(i)}{\sum_{i=1}^{N_c} h_{tot}(i)}$, for the distribution of descriptors among the correspondence classes before the split and similarly for p_1, \dots, p_k . The entropy before the split is defined as $E_{tot} = \sum_{i=1}^{N_c} -p_{tot}(i) \log_2(p_{tot}(i))$, and similarly for the k branches, $E_j = \sum_{i=1}^{N_c} -p_j(i) \log_2(p_j(i))$. For the split as a whole we define the entropy as $E_{split} = \sum_{j=1}^k \frac{n_j}{n_{tot}} E_j$. Here $n_j = \sum_{i=1}^{N_c} h_j(i)$. Ideally each split, which uses $\log_2(k)$ extra bits of information, should lower the entropy with $\log_2(k)$ bits, i.e. we expect E_{split} to be approximately $\log_2(k)$ less than E_{tot} . In practice it is difficult to split all examples in the descriptor space as cleanly as this.

4.2 Optimizing Entropy

For training data $(x_1, \dots, x_N), (c_1, \dots, c_N)$ with possible weights (y_1, \dots, y_N) , it is thus possible to define the split entropy E_{split} as a function of the division parameters z . For hard assignment, using $z = (c_1, \dots, c_k)$, this function is not smooth. The entropy is typically constant as the decision boundaries are perturbed as long as they do not pass through any of the points x_i . For soft assignment, however, entropy is a smooth function of the division parameters $z = (c_1, \dots, c_k, m)$.

In our experiments we have tried a few different approaches for optimizing E with respect to z . We did not optimize E with respect to the margin m in this paper.

In the main approach we initialize using k-means iterations with a couple of different starting points. The best initial estimate is then used as an initial estimate to a non-linear optimization of z . Here we have calculated the analytical derivatives $\frac{dE}{dz}$, which are then used in a non linear optimization.

The entropy for the split can be written as $E_{split} = \sum_{j=1}^k \frac{n_j}{n_{tot}} E_j$. which since $n_j p_j(i) = h_j(i)$ gives $E_{split} = \sum_{j=1}^k \frac{1}{n_{tot}} \sum_{i=1}^{N_c} (-h_j(i) \log_2(p_j(i)))$. The derivative of E_{split} is thus

$$\frac{dE_{split}}{dz} = \frac{-1}{n_{tot}} \sum_{j=1}^k \sum_{i=1}^{N_c} \left(\frac{dh_j(i)}{dz} \log_2(p_j(i)) + \frac{n_j}{\ln(2)} \frac{dp_j(i)}{dz} \right) \tag{2}$$

Here the sum of the second term over all i is zero, since the sum of the probabilities is constant. Thus

$$\frac{dE_{split}}{dz} = \frac{1}{n_{tot}} \sum_{j=1}^k \sum_{i=1}^{N_c} \left(-\frac{dh_j(i)}{dz} \log_2(p_j(i)) \right). \tag{3}$$

Here

$$\frac{dp_j(i)}{dz} = \frac{1}{n_j} \frac{dh_j(i)}{dz} - \frac{h_j(i)}{n_j^2} \sum_{m=1}^{N_c} \frac{dh_j(m)}{dz}. \tag{4}$$

The derivatives of the histogram bins are $\frac{dh_j(i)}{dz} = \sum_{j,c_j=i} \frac{d\omega_j(j)}{dz}$. Finally the derivatives of the weights are

$$\frac{d\omega_j(i)}{dz} = \frac{\frac{de_j(i)}{dz}}{\sum_{m=1}^k e_m(i)} - e_j(i) \frac{\sum_{m=1}^k \frac{de_m(i)}{dz}}{\left(\sum_{m=1}^k e_m(i)\right)^2}, \tag{5}$$

where

$$\frac{de_j(i)}{dz} = e_j(i) \left(\frac{(x_i - c_j)}{m|x_i - c_j|} \frac{dc_j}{dz} - \frac{|x_i - c_j|}{m^2} \frac{dm}{dz} \right). \tag{6}$$

The value E and the gradient $\frac{dE}{dz}$ are utilized in a non-linear optimization update with the limited-memory Broyden-Fletcher-Goldfarb-Shanno method, [3, 12]. In the implementation we have limited the maximum number of iterations of the optimization to 20 iterations for the first levels, but increased to 30 iterations for the subsequent levels to avoid over-fitting. This scheme is general for different values of k .

5 Experimental Validation

We have tested our method on vocabulary construction with real image data. The dataset is described in details in Section 5.1. The resulting vocabularies are evaluated in Section 5.2.

5.1 Dataset and Evaluation

We use three sets of data with partial ground truth on correspondences, from the UBC Patch Data [25]. These datasets contain scale and orientation normalized patches (from either difference of Gaussians (DOG) or Harris corners detectors) sampled from 3D reconstructions of three landmarks (Statue of Liberty, Notre Dame and Yosemite). In Figure 3 we show two sets of patches in the same correspondence class from the Statue of Liberty and Notredame dataset respectively. Each dataset (Notre Dame, Liberty and Yosemite) contain approximately 500K descriptors in 150K correspondence classes.

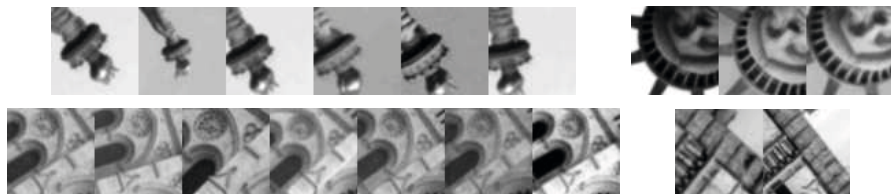


Fig. 3. Correspondence patches from the Statue of Liberty (Top) and Notredame (Bottom) dataset

For our experiments, we extracted SIFT descriptors on DOG patches. To provide correspondence ground truth for training and evaluation, we generated the whole set of matched pairs for each correspondence class, and a random non-matched for each patch to form non-matched pairs (with the same reference image as suggested by [25]).

We then used the methods in Section 4 to construct vocabularies based on the SIFT descriptors and partial ground truth for these datasets. We have here used a subset of the data for the training and another non-overlapping subset for the testing.

5.2 Vocabularies with Hard Assignment

In the first experiment we trained vocabularies with hierarchical $k = 3$ splits with 9 levels by optimizing entropy based on soft assignment. When testing, we used hard assignment with respect to the optimized k cluster centers. We compare the results with those of hierarchical k -means with 3 splits in each node. The vocabularies are trained both for hierarchical k -means and for entropy optimization on a subset (50 percent) of the Statue of Liberty dataset.

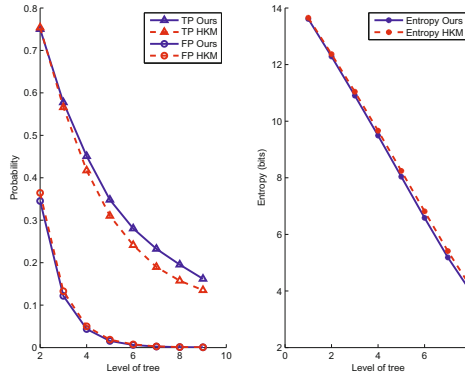


Fig. 4. Evaluation on Liberty data. (50% for training and 20% for testing with $k = 3$. Left: Estimated probability of two corresponding (TPR) and two random descriptors (FPR) ending up in the same word as a function of tree depth. Middle : Entropy as a function of tree depth. Notice that with each depth entropy is lowered close to 1.5 bit.

The resulting vocabularies were then tested on a subset of the Statue of Liberty dataset (20 percent) which does not contain the same correspondence classes as were used in the training. We measured how the entropy decreases with increasing vocabulary size. Also a subset of matching points were used to test how often two matching points (True Positive rate, TPR) end up in the same word. Finally a subset of pair of random unmatched points in the dataset were used to see how often two unmatched points end up in the same word (False Positive rate, FPR). This result is shown in figure 4. Notice also that the probability of two matching features ending up in the same word is higher for the entropy minimized vocabulary for unseen data points, which suggests the generality of the learned vocabulary. Moreover, we obtained slightly lower FPR across different levels of the tree. We also observed that the entropy is lowered by approximately 1.5 bits ($\log_2(3) \approx 1.585$) with each level in the hierarchical split, but slightly more so when using an entropy minimized vocabulary, suggesting that entropy is a fair measure on the quality of the resulting clusters. In this experiment we used a fixed setting for the margin $m = 1$.

To further investigate the generality of the method, we have trained vocabularies on 50% of the features from the Statue of Liberty, the Notre Dame and the Yosemite datasets and tested it on the remaining 50% features. The optimized vocabulary compared to hierarchical k-means results in lower entropy and higher TPR. The resulting plot is very similar to Figure 4 suggesting the optimized vocabulary generalized well to new data.

5.3 Vocabularies with Soft Assignment

In the next experiment, we used the same vocabulary as in Section 5.2, but switched to soft assignment when passing unseen feature points down the

hierarchical tree. Features can then fall in several children nodes where their weights to the corresponding centers are larger than a preset threshold $\epsilon = 10^{-6}$. This results in multiple word ID's for a single feature. If we regard two features as matched if they share the same words as before, we will expect higher TPR as matched features will have greater possibility of overlapping. On the other hand, two random non-matched features will also tend to have one of the word ID's in common. Consequently, the FPR will also increase. Here, we also fixed the margin to $m = 1$ during training.

We expect our optimization framework to improve the TPR while controlling the FPR by training on ground truth data. In Figure 4 we can see that, the proposed method is marginally better than the hierarchical k-means with respect to the TPR and FPR curve. Only achieving marginal optimality might be due to the fact that we have not used enough data for training. On the other hand, we noted that both soft assignment vocabularies have better matching property than hard assignment vocabularies. For instance if we aim for 5% false positive rate, soft assignment achieves approximately 60% true positive rate while hard assignment obtains only 45%.

5.4 Effects of Margin

In this section, we studied the effect of different margins on soft assignment tests. Here we fixed the value of m during the training stage and evaluate how margins affect the match performance for test data. Note that as m becomes smaller, the soft assignment behaves in a similar way as hard assignment. On the other hand, larger m implies more ambiguities for each features ending up in different words; therefore, possibly higher false positive rate for matching.

We have experimented with $m = 0.25, 0.5, 1$ (Figure 5). As expected, when increasing the margin we can achieve better TPR with the trade-off of worse FPR at the same level of the tree. The optimized vocabularies are better than hierarchical k-means across different margins indicating the usefulness of utilizing ground truth. More importantly, the overall statistics shed lights on how we should choose the size of the vocabularies (level of hierarchical trees). The converging trend of all curves with different m 's suggests that at certain number of words, we can always obtain better TPR with soft assignment but approximately the same FPR. However, such better performance comes at the price of heavier computation when assigning features to multiple leaf nodes. If m is too large, features will end up in many words at the leaf node. Therefore, the efficiency of vocabulary representation of features is overwhelmed by computing the intersections in the space of word ID's.

5.5 Image Retrieval

In this section, we verify the usefulness of optimized vocabulary in the recognition pipeline on the Oxford 5K dataset [20, 21]. The task is to retrieve similar images to the 55 query images (5 for each of the 11 landmarks in Oxford) in the dataset of 5062 images. The performance is then evaluate with mean Average

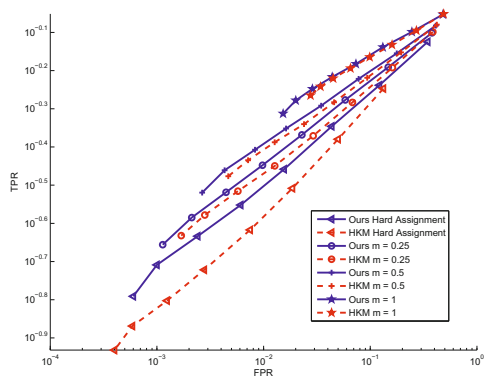


Fig. 5. The effects of different margins on soft assignment with respect to TPR and FPR. $m = 0.25, 0.5, 1$ and hard assignment, where $k = 3$.

Table 1. mAPs with different levels of hierarchical k-means and our method with $k = 3$ on the Oxford 5K dataset

Level	HIK $k = 3$	Our Method $k = 3$
9	0.1744	0.1955
10	0.1849	0.1979
11	0.1805	0.1837

Precision (mAP) score. Higher mAP indicates that the underlying system on average retrieves the similar corresponding images at the top of the ranked list.

We follow the BOW baseline system, and use a hierarchical k-means vocabulary and our optimized vocabulary respectively for vocabulary training. We trained the vocabulary with 50% of a mixture of Liberty, Notredame and Yosemite patch data which contains approximately 800K features and 250K correspondence classes in total. After that, we use hard assignment to quantize the SIFT features from the Oxford 5K images. We observe that our optimized vocabulary is always superior to the unsupervised hierarchical k-means by capturing the local characteristics of the feature space. When increasing the number of levels to 11 we can see that the performance drops both for hierarchical k-means and our method. This can be an indication that the vocabulary is over-trained on the patch data. Note that these results are not directly comparable with [20] in which vocabularies are trained on features in the images where the actual retrieval is performed.

6 Conclusions

In this paper, we have developed a general method for optimizing hierarchical visual vocabularies using correspondence ground truth between features. The ground truth prior knowledge on the feature space is utilized to refine the local

structures of the trained vocabulary such that matched features will tend to fall in the same word. We propose the use of a soft margin hierarchical k-splits tree where the optimization of the tree is based on minimizing an entropy criterion defined on ground truth data. Unlike the traditional clustering methods such as hierarchical k-means, optimization with respect to entropy enables the cluster centers to adjust locally to capture the implicit connections between features. We demonstrate the method on real dataset with promising results. Compared to the unsupervised hierarchical k-means with hard assignment, the optimized vocabulary obtained higher true positive rate and lower false positive rates. We also show that soft assignment boosts the overall performance regarding matching features.

We have in this paper focused on the optimization aspects of vocabulary training using existing ground truth data. Due to the high dimensionality of the parameter space, the learning requires huge amounts of data in order to avoid over-fitting. Therefore, as future work, we aim to generate and utilize large scale ground truth data to facilitate robust training with geometry or deformable models. We need also to cope with the inherent quantization errors introduced by hierarchical quantization. We would like to investigate how the soft-assignment process might mitigate the such quantization errors. To enable large scale training, we are also pursuing efficient optimization techniques for our approach.

Acknowledgement. We would like to thank the Matthew Brown, Gang Hua and Simon Winder for making their UBC Patch Dataset available. The research leading to these results has received funding from the strategic research projects ELLIIT and eSENCE, Swedish Governmental Agency for Innovation Systems project AUTOMETa, and Swedish Foundation for Strategic Research projects ENGR0SS and VINST.

References

1. Agarwal, S., Snavely, N., Simon, I., Seitz, S.M., Szeliski, R.: Building rome in a day. In: Proc. 12th Int. Conf. on Computer Vision, Kyoto, Japan (2009)
2. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
3. Broyden, C.G.: The convergence of a class of double-rank minimization algorithms. *Journal of the Institute of Mathematics and its Applications* 6, 76–90 (1970)
4. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology (2007)
5. Hartley, R.I., Zisserman, A.: *Multiple View Geometry in Computer Vision*, 2nd edn. Cambridge University Press, Cambridge (2004)
6. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proc. of the 4th Alvey Vision Conference, pp. 147–151 (1988)
7. Ji, R., Yao, H., Sun, X., Zhong, B., Gao, W.: Towards semantic embedding in visual vocabulary. In: Proc. Conf. Computer Vision and Pattern Recognition, San Francisco, California, USA (2010)

8. Karlsson, J., Åström, K.: MDL patch correspondences on unlabeled data. In: Proc. International Conference on Pattern Recognition, Tampa, USA (2008)
9. Lamrous, S., Taïleb, M.: Divisive hierarchical k-means. In: CIMCA-IAWTIC 2006. IEEE Computer Society Press, Los Alamitos (2006)
10. Lazebnik, S., Raginsky, M.: Supervised learning of quantizer codebooks by information loss minimization. *IEEE Trans. Pattern Analysis and Machine Intelligence* 31, 1294–1309 (2009)
11. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision* 60, 91–110 (2004)
12. Luenberger, D.G.: *Linear and Nonlinear Programming*. Addison-Wesley, Reading (1984)
13. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. *Image Vision Computing* 22, 761–767 (2004)
14. Mikolajczyk, K., Schmid, C.: An affine invariant interest point detector. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) *ECCV 2002*. LNCS, vol. 2350, pp. 128–142. Springer, Heidelberg (2002)
15. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. In: *Proc. Conf. Computer Vision and Pattern Recognition* (2003)
16. Mikulik, A., Perdoch, M., Chum, O., Matas, J.: Learning a fine vocabulary. In: *Proc. 11th European Conf. on Computer Vision*, Crete, Greece (2010)
17. Moosmann, F., Triggs, B., Jurie, F.: Randomized clustering forests for building fast and discriminative visual vocabularies. In: *Twentieth Annual Conference on Neural Information Processing Systems*, Vancouver, Canada (2006)
18. Nistér, D., Stewénius, H.: Scalable recognition with a vocabulary tree. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2161–2168 (2006)
19. Perronnin, F.: Universal and adapted vocabularies for generic visual categorization. *IEEE Trans. Pattern Analysis and Machine Intelligence* 30, 1243–1256 (2008)
20. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2007)
21. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2008)
22. Philbin, J., Isard, M., Sivic, J., Zisserman, A.: Descriptor learning for efficient retrieval. In: *Proc. 11th European Conf. on Computer Vision*, Crete, Greece (2010)
23. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: *Proceedings of the International Conference on Computer Vision* (2003)
24. Strecha, C., Bronstein, A., Bronstein, M., Fua, P.: LDAhash: Improved matching with smaller descriptors. Technical report, CVlab, EPFL Switzerland, Tel-Aviv University and Israel Institute of Technology, Israel (2010)
25. Winder, S., Hua, G., Brown, M.: Picking the best daisy. In: *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, Miami (2009)
26. Winn, J., Criminisi, T., Minka, T.: Object categorization by learned visual dictionary. In: *Proc. 10th Int. Conf. on Computer Vision*, Beijing, China (2005)

Totally-Corrective Multi-class Boosting

Zhihui Hao¹, Chunhua Shen^{2,3}, Nick Barnes^{2,3}, and Bo Wang¹

¹ Beijing Institute of Technology, Beijing 100081, China

² NICTA, Canberra Research Laboratory, Canberra, ACT 2601, Australia

³ Australian National University, Canberra, ACT 0200, Australia

hzhbit@gmail.com

Abstract. We proffer totally-corrective multi-class boosting algorithms in this work. First, we discuss the methods that extend two-class boosting to multi-class case by studying two existing boosting algorithms: AdaBoost.MO and SAMME, and formulate convex optimization problems that minimize their regularized cost functions. Then we propose a column-generation based totally-corrective framework for multi-class boosting learning by looking at the Lagrange dual problems. Experimental results on UCI datasets show that the new algorithms have comparable generalization capability but converge much faster than their counterparts. Experiments on MNIST handwriting digit classification also demonstrate the effectiveness of the proposed algorithms.

1 Introduction

Boosting is a learning method to train a strong classifier by combining many weak hypotheses. The most popular boosting algorithm is AdaBoost, proposed by Freund and Schapire [1]. As the first practical boosting algorithm, AdaBoost has been applied in many tasks, such as face detection [2] and image retrieval [3]. Most well studied boosting algorithms are designed for two-class classification problems, to separate positive instances from negative instances.

Typically, weak hypotheses in boosting learning are required to generate a training error lower than $1/2$, in order to receive a nonnegative coefficient. In binary case, this is equivalent to say that any classifier better than random guessing is acceptable to be a weak learner. However, this requirement becomes harder in multi-class case, where random guessing only has an accuracy of $1/K$, if K classes in all are included. For this reason, boosting algorithms often fail at directly applying to multi-class problems.

A natural idea to overcome the difficulty is to reduce them into multiple binary ones, since the great success of two-class boosting algorithms. AdaBoost.MO [4] achieves this purpose by introducing a coding matrix. The final classifier is multi-dimensional, with each entry boosted on a relabeled set of training data. Algorithms that use this strategy include -MO, -OC [5], -ECC [6], *etc.*

Recently, Zhu *et al.* [7] proposed a new extension called SAMME, which directly solves the multi-class problem without decomposing. One single multi-class hypothesis is trained at each iteration. Compared with -MO, SAMME

is conceptually simpler and easier to implement. As reported in [7], SAMME performs similarly to those previous multi-class boosting.

From the perspective of optimization, what AdaBoost.MO and SAMME try to solve are both convex optimization problems. In [8], Shen and Li indicated that with the help of the column generation technique, any convex loss function problem might be optimized through optimizing its corresponding dual in a boosting fashion. They explored two-class boosting algorithms including AdaBoost, LogitBoost [9] and LPBoost [10]. Inspired by their work, we derive the Lagrange duals of multi-class boosting algorithms including AdaBoost.MO and SAMME. Then we design a totally corrective boosting framework for multi-class classification problems.

The paper is organized as follows. First we briefly describe AdaBoost.MO and SAMME in Section 2, then we derive the Lagrange duals and present our boosting learning in Section 3. Finally we compare all these multi-class algorithms and show the experimental results in Section 4.

2 Multi-class Boosting

Let us introduce some notation before we proceed. The multi-class classification training data are given by (\mathbf{x}_i, y_i) , $i = 1 \dots N$. Here \mathbf{x}_i is a pattern and y_i is the label, which takes a value from the space $\mathbb{Y} = \{1, \dots, K\}$ if we have K classes. The goal of multi-class boosting is then to find a classifier $\mathbf{f} : \mathbb{X} \rightarrow \mathbb{Y}$ which assigns one and only one label to a new instance (\mathbf{x}, y) with a minimum probability of $\mathbf{f}(\mathbf{x}) \neq y$.

For the sake of being self-contained, we briefly review the multi-class boosting algorithms AdaBoost.MO and SAMME in this section.

2.1 AdaBoost.MO

Before boosting, AdaBoost.MO encodes each label into a vector by introducing a coding matrix. The matrix \mathbf{M} could be constructed by ECOC [11] or random codes [6]. In this paper, our conclusions will not be affected by the coding

Algorithm 1. AdaBoost.MO (Schapire & Singer, 1999)

Given training data (\mathbf{x}_i, y_i) , $i = 1 \dots N$.

(1) Initialize the weights $D_1(i, k) = 1/(NK)$, $i = 1 \dots N$, $k = 1 \dots K$;

for $t = 1$ to T **do**

(2) Normalize D_t ;

(3) Train K weak classifiers $h_k^{(t)}$, $k = 1 \dots K$ using distribution D_t ;

(4) Compute $w^{(t)}$;

(5) Update weights $D_{t+1}(i, k) = D_t(i, k) \exp\left(-w^{(t)} \lambda_k(y_i) h_k^{(t)}(\mathbf{x}_i)\right)$;

end for

Output $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_K(\mathbf{x})]^\top$.

strategy, therefore, for ease of exposition, we can consider the simplest case. Given a one-to-one mapping

$$\lambda_k(y) = \begin{cases} 1 & k = y, \\ -1 & k \neq y; \end{cases}$$

the vector of label y will be $\mathbf{y}^{K \times 1} = [-1, \dots, 1, \dots, -1]^\top$, that is, only one element in the vector is +1 which represents the true label, and the others are all -1. This is usually called a one-per-class approach. In this case, the coding matrix \mathbf{M} is similar to a K -by- K identity matrix. Then the training labels turn to be a label matrix, and each column of it defines a binary partition over training samples, on which a binary classifier is feasible to be trained. Apparently, the output of AdaBoost.MO is a K -dimensional classifier $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_K(\mathbf{x})]^\top$, with each sub-classifier boosted from a set of weak hypotheses $f_k(\mathbf{x}) = \sum_{j=1}^T w^{(j)} h_k^{(j)}$, $k = 1 \dots K$. The algorithm is briefly summarized in Algorithm 1.

For a newly observed instance \mathbf{x}^* , the label y^* is predicted by decoding the output with some strategy, such as minimizing the loss:

$$y^* = \operatorname{argmin}_{y \in \mathbb{Y}} \sum_{k=1}^K \exp(\lambda_k(y) f_k(\mathbf{x})).$$

AdaBoost.MO has been proved to perform a stage-wise gradient descent procedure and minimize an exponential function [12]:

$$\text{Loss} = \sum_{i,k} \exp(-\lambda_k(y_i) f_k(x_i)). \tag{1}$$

2.2 SAMME

SAMME adopts a multi-class exponential loss function and solves a multi-class problem without reducing it into binary ones. The output coding strategy in SAMME is different to the one used in AdaBoost.MO. Here for a label y , the corresponding vector is $\mathbf{y} = [y_1, \dots, y_K]^\top$, where its k -th element is

$$y_k = \begin{cases} 1 & \text{if } k = y, \\ -\frac{1}{K-1} & \text{otherwise.} \end{cases}$$

Under such a coding strategy, the multi-class loss function can be expressed as

$$\begin{aligned} \text{Loss}(\mathbf{y}, \mathbf{f}(\mathbf{x})) &= \exp\left(-\frac{1}{K} (y_1 f_1(\mathbf{x}) + \dots + y_K f_K(\mathbf{x}))\right) \\ &= \exp\left(-\frac{1}{K} \mathbf{y}^\top \mathbf{f}(\mathbf{x})\right). \end{aligned} \tag{2}$$

The process of SAMME is very similar to AdaBoost. The major difference is that SAMME adds a new term of $\log(K - 1)$ to computing $w^{(t)}$ at each iteration, and thus the underlying weak learner is only required to be slightly better than random guessing, that is, more than $1/K$ accuracy.

The algorithm of SAMME eventually produces a linear combination of multi-class weak hypotheses:

$$\mathbf{f}(\mathbf{x}) = \sum_{j=1}^T w^{(j)} \mathbf{h}^{(j)}(\mathbf{x}),$$

with $\mathbf{h}^{(j)}(\cdot) = [h_1^{(j)}(\cdot), \dots, h_K^{(j)}(\cdot)]^\top$. It is easy to observe that for assembled classifier $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_K(\mathbf{x})]^\top$ with each entry $f_k = \sum_j w^{(j)} h_k^{(j)}$, we have

$$f_1(\mathbf{x}) + f_2(\mathbf{x}) + \dots + f_K(\mathbf{x}) = 0. \tag{3}$$

3 Totally Corrective Multi-class Boosting

The cost functions of AdaBoost.MO and SAMME are both convex. Thus we are able to derive the corresponding Lagrange duals and look at multi-class problems from a different view. Based on the duals, we design new algorithms for boosting learning. We call the two totally corrective methods that are based on the cost functions of AdaBoost.MO and SAMME as MultiBoost_{TC1} and MultiBoost_{TC2}, respectively.

3.1 MultiBoost_{TC1}

We formally rewrite the problem (II) that AdaBoost.MO optimizes:

$$\begin{aligned} \min_{\mathbf{w}} \sum_{i,k} \exp \left(- \sum_{j=1}^T w^{(j)} \lambda_k(y_i) h_k^{(j)}(\mathbf{x}_i) \right) \\ \text{s.t. } \mathbf{w} \succeq 0, \|\mathbf{w}\|_1 \leq \theta. \end{aligned} \tag{4}$$

This is a convex program in \mathbf{w} . Note that the constraint $\|\mathbf{w}\|_1 \leq \theta$ is not explicitly enforced in the AdaBoost.MO algorithm. However, without this regularization constraint, one can always make the cost function approach zero via enlarging the solution \mathbf{w} by an arbitrarily large factor. Moreover, it is easy to check that for a convex and monotonically increasing loss function, $\|\mathbf{w}\|_1 \leq \theta$ is equivalent to $\|\mathbf{w}\|_1 = \theta$. In other words, \mathbf{w} always locates at the boundary of the feasibility set.

Theorem 1. *The Lagrange dual problem of (4) is*

$$\begin{aligned} \max_{r, \mathbf{u}} -r\theta - \sum_{i,k} u_{i,k} \log u_{i,k} + \sum_{i,k} u_{i,k} \\ \text{s.t. } \sum_{i,k} u_{i,k} \lambda_k(y_i) [h_k^{(1)}(\mathbf{x}_i) \dots h_k^{(T)}(\mathbf{x}_i)] < r \mathbf{1}^\top, \mathbf{u} \succeq 0 \end{aligned} \tag{5}$$

Proof. The proof is provided in the Appendix.

The number of underlying weak classifiers may be infinitely large. In order to solve the dual problem, we use the column generation technique [10] to add one constraint at a time until an optimal solution is identified. That is, at each iteration, we find the weak classifier that most violates the constraint in the dual problem. So at time t , such an optimal multi-dimensional classifier $\mathbf{h}^{(t)}(\cdot) = [h_1^{(t)}(\cdot), \dots, h_K^{(t)}(\cdot)]^\top$ can be found by

$$\mathbf{h}^{(t)}(\cdot) = \operatorname{argmax}_{\mathbf{h}(\cdot)} \sum_{i,k} u_{i,k} \lambda_k(y_i) h_k(\mathbf{x}_i), \tag{6}$$

which is equivalent to solving

$$h_k^{(t)}(\cdot) = \operatorname{argmax}_{h_k(\cdot)} \sum_{i=1}^N u_{i,k} \lambda_k(y_i) h_k(\mathbf{x}_i), \forall k = 1 \dots K. \tag{7}$$

This is exactly the same as the strategy that AdaBoost.MO adopts for generating the weak classifier at each iteration, that is, to find the weak classifier that produces the minimum weighted training error.

3.2 MultiBoost_{TC2}

In this section, we present another boosting algorithm based on the multi-class exponential loss function, which has been used in SAMME:

$$\begin{aligned} \text{Loss} &= \sum_{i=1}^N \exp\left(-\frac{1}{K} \mathbf{y}_i^\top \mathbf{f}(\mathbf{x}_i)\right) \\ &= \sum_{i=1}^N \exp\left(-\frac{1}{K} \mathbf{y}_i^\top \sum_{j=1}^T w^{(j)} \mathbf{h}^{(j)}(\mathbf{x}_i)\right) \end{aligned}$$

Thus the problem that we want to solve is

$$\begin{aligned} \min_{\mathbf{w}} \quad & \sum_{i=1}^N \exp\left(-\frac{1}{K} \sum_j w^{(j)} \mathbf{y}_i^\top \mathbf{h}^{(j)}(\mathbf{x}_i)\right) \\ \text{s.t.} \quad & \mathbf{w} \succeq 0, \|\mathbf{w}\|_1 \leq \theta. \end{aligned} \tag{8}$$

It is easy to verify that the above is also a convex problem, and $\|\mathbf{w}\|_1 \leq \theta$ is equivalent to $\|\mathbf{w}\|_1 = \theta$.

Theorem 2. *The Lagrange dual problem of (8) is*

$$\begin{aligned} \max_{r, \mathbf{u}} \quad & -\theta r - \sum_{i=1}^N u_i \log u_i + \sum_{i=1}^N u_i \\ \text{s.t.} \quad & \frac{1}{K} \sum_{i=1}^N u_i \mathbf{y}_i^\top [\mathbf{h}^{(1)}(\mathbf{x}_i) \dots \mathbf{h}^{(T)}(\mathbf{x}_i)] \leq r \mathbf{1}^\top. \end{aligned} \tag{9}$$

For a detailed derivation, we refer the reader to the proof of Theorem 11.

Using the idea of column generation, we find the weak classifier that most violates the constraint in the dual problem. So at each iteration such an optimal weak classifier can be found by

$$\mathbf{h}^*(\cdot) = \underset{\mathbf{h}(\cdot)}{\operatorname{argmax}} \sum_{i=1}^N u_i \mathbf{y}_i^\top \mathbf{h}(\mathbf{x}_i). \tag{10}$$

Recall that $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), \dots, h_K(\mathbf{x})]^\top$ corresponds to a multi-class classifier $g(\mathbf{x})$ such that

$$h_k(\mathbf{x}) = 1, \text{ if } g(\mathbf{x}) = k. \tag{11}$$

In other words, $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), \dots, h_K(\mathbf{x})]^\top$ is obtained by $h_{g(\mathbf{x})}(\mathbf{x}) = 1$ and for any other k , $h_k(\mathbf{x}) = -\frac{1}{K-1}$.

Theorem 3. *Problem (10) is equivalent to finding the weak classifier $g(\mathbf{x})$ such that the weighted classification error of $g(\mathbf{x})$ is minimal; i.e., same as in SAMME.*

Proof. Here we provide a simpler explanation that differs to the one given in [7]. Mathematically, the above theorem says that (10) is equivalent to solving

$$g^*(\mathbf{x}) = \underset{g(\mathbf{x})}{\operatorname{argmin}} \sum_{i: g(\mathbf{x}_i) \neq y_i} u_i. \tag{12}$$

We know that $\mathbf{y}_i \in \mathbb{R}^K$ is a vector with its y_i -th entry equal to 1 and all the others are $-\frac{1}{K-1}$, and $\mathbf{h}(\mathbf{x}_i)$ is a vector of the same format. Therefore the inner product $\mathbf{y}_i^\top \mathbf{h}(\mathbf{x}_i)$ can only take values from a binary set:

$$\mathbf{y}_i^\top \mathbf{h}(\mathbf{x}_i) = \begin{cases} \frac{K}{K-1} & \text{if } g(\mathbf{x}_i) = y_i \text{ (correctly classified)} \\ -\frac{K}{(K-1)^2} & \text{otherwise (wrongly classified).} \end{cases}$$

Hence,

$$\begin{aligned} \sum_{i=1}^N u_i \mathbf{y}_i^\top \mathbf{h}(\mathbf{x}_i) &= \frac{K}{K-1} \left(\sum_{i: g(\mathbf{x}_i) = y_i} u_i - \frac{1}{K-1} \sum_{i: g(\mathbf{x}_i) \neq y_i} u_i \right) \\ &= \frac{K}{K-1} \left(\sum_{i=1}^N u_i - \frac{K}{K-1} \sum_{i: g(\mathbf{x}_i) \neq y_i} u_i \right). \end{aligned} \tag{13}$$

Because $\sum_i u_i$ does not depend on the weak classifier to be chosen, (10) is equivalent to (12).

Next we need to find the connection between the primal variables \mathbf{w} and dual variables \mathbf{u} and r . Take MultiBoost_{TC2} for example, strong duality holds between the primal (8) and dual (9) [13]. According to the KKT conditions [13], we have

$$u_i^* = \exp \gamma_i^* = \exp \left(-\frac{1}{K} \sum_j w^{*(j)} \mathbf{y}_i^\top \mathbf{h}^{(j)}(\mathbf{x}_i) \right). \tag{14}$$

Algorithm 2. Totally Corrective Multi-class Boosting

Given training data $(\mathbf{x}_i, \mathbf{y}_i)$, $i = 1 \dots N$; termination threshold $\epsilon > 0$; regularization coefficient θ ; maximum training steps T .

- (1) Initialize $\mathbf{w} = 0$; $t = 0$;
 (MultiBoost_{TC1}) $u_{i,k} = 1/(NK)$, $\forall i = 1 \dots N, \forall k = 1 \dots K$.
 (MultiBoost_{TC2}) $u_i = 1/N$, $\forall i = 1 \dots N$.

while true **do**

- (2) Find a new weak classifier $\mathbf{h}^*(\cdot)$ with minimum weighted training error, that is, solve the problem (7) or (10);
- (3) Check for optimal solution: if
 (MultiBoost_{TC1}) $\sum_{i,k} u_{i,k} \lambda_k(\mathbf{y}_i) \mathbf{h}_k^*(\mathbf{x}_i) < r + \epsilon$.
 (MultiBoost_{TC2}) $\frac{1}{K} \sum_{i=1}^N u_i \mathbf{y}_i^\top \mathbf{h}^*(\mathbf{x}_i) < r + \epsilon$
 then break;
- (4) Add new constraint to the corresponding dual problem;
- (5) Solve the dual (5) or (9);
- (6) $t = t + 1$; if $t > T$, then break;

end while

- (7) Calculate the primal \mathbf{w} according to the solutions of dual and KKT condition.
 - (8) Output $\mathbf{f}(\cdot) = \sum_{j=1}^t w^{(j)} \mathbf{h}^{(j)}(\cdot)$.
-

r^* can be recovered from \mathbf{u}^* by

$$r^* = \max_{j=1, \dots, T} \left\{ \frac{1}{K} \sum_{i=1}^N u_i \mathbf{y}_i^\top \mathbf{h}^{(j)}(\mathbf{x}_i) \right\}. \tag{15}$$

This is because at optimality, at least one of dual problem’s constraints is strictly equal. In our experiments, we have used MOSEK [14], which is a primal-dual interior-point solver. Both the primal and dual solutions are given at convergence by MOSEK.

In both MultiBoost_{TC1} and MultiBoost_{TC2}, all the weak classifiers are updated at each iteration. In this sense, both of them are totally corrective [15]. Looking at the dual programs of these two algorithms, they are quite similar, thus we can summarize them in Algorithm 2 as a totally-corrective multi-class boosting framework.

4 Experiments

In this section, we describe two sets of experiments that we ran to verify our algorithms. The first set of experiments compares the algorithms on a collection of datasets from the UCI Irvine machine learning repository. The second experiment makes a comparison on a real image dataset. The algorithms we compare include AdaBoost.MO, MultiBoost_{TC1}, SAMME and MultiBoost_{TC2}. The dual optimization problems within are solved by using the off-the-shelf MOSEK package [14].

4.1 UCI Datasets

We have collected 15 datasets from UCI repository to run the first experiment. For each time, we randomly select 70% samples for training and 30% for test. Samples from the same class are partitioned in proportion to maintain the balance of multi-class problems. This procedure is repeated ten times and the results are finally averaged.

MultiBoost_{TC1}. The only parameter to be tuned in MultiBoost_{TC1} is the regularization coefficient θ . We choose it by a two-step cross-validation scheme. First we run a five-fold cross-validation on a set of sparse and uniformly distributed values $\{10, 20, 30, \dots, 100, 200, \dots, 1000\}$. According to the results, one-fifth of the candidates are retained. Then we expand them into a new pool to fine tune the parameter. This scheme seems to find a better parameter value.

Due to the simplicity, we use decision stumps as weak classifiers of AdaBoost.MO and MultiBoost_{TC1}. The experimental results are shown in Table 1. The maximum numbers of training steps are set to be 100, 500 and 1000. The table reports the test error and the number of iterations when the training error converges to zero. For the cases where the value is very close to the maximum, for example AdaBoost.MO's convergence steps on dataset *Vehicle*, the training actually did not converge within the given number of iterations. Looking at the results of AdaBoost.MO and MultiBoost_{TC1}, we have the following conclusions.

1. The convergence speed of MultiBoost_{TC1} is faster than AdaBoost.MO in most cases. This is because AdaBoost.MO performs a slow gradient descent process while MultiBoost_{TC1} is totally corrective. This means a classifier with less weak hypotheses can be obtained. A simpler model, for example, a cascaded face detector of smaller size can speed up the detection process, which is critical to many applications, especially those with the real-time requirements.

2. Note that from the test error results in Table 1, it seems that the stage-wise boosting algorithms are slightly better than the proposed totally-corrective algorithms on most of the nine tested datasets. However, the difference is negligible. Indeed, statistical testing does not show a significant performance difference between the proposed algorithm and its stage-wise counterpart. We conjecture that if we carefully tune the regularization parameter, our algorithm's performance could be improved.

The conclusions are consistent with [8], where they compared AdaBoost and its totally-corrective version AdaBoost-CG. This meets our expectation since MultiBoost_{TC1} can be regarded as a simple extension of AdaBoost-CG to multi-class case.

MultiBoost_{TC2}. To use column generation technique in dual problems, we have to find a weak classifier that minimizes the weighted training error for every iteration. To design a multi-class classifier we may consider using decision trees, such as classification and regression tree (CART), however, a decision tree that minimizes the training error is necessarily a fully grown tree which eliminates any training error via iterative splitting. This perfect tree stops the boosting

Table 1. Test errors and iterations when training errors converge. All tests are run 10 times and the results are averaged. AdaBoost.MO(abbreviated to MO) and MultiBoost_{TC1}(TC1) use decision stumps as weak learners. SAMME and MultiBoost_{TC2}(TC2) use terminal node bounded CARTs as weak learners.

dataset	algorithm	test error 100	test error 500	test error 1000	#conv. 100	#conv. 500	#conv. 1000
Segmentation	MO	0.081±0.011	0.081±0.009	0.078±0.010	40.4±5.2	40.4±5.2	45.3±7.4
	TC1	0.108±0.013	0.119±0.017	0.114±0.013	16.8±1.5	15.4±2.7	15.7±2.3
	SAMME	0.060±0.014	0.055±0.012	0.056±0.010	4.8±1.4	3.0±0.0	3.6±1.8
	TC2	0.064±0.012	0.065±0.009	0.062±0.010	4.1±0.8	3.4±0.5	3.4±0.5
Thyroid	MO	0.006±0.001	0.006±0.001	0.006±0.001	98.8±1.5	292.4±60.2	270.2±52.1
	TC1	0.006±0.001	0.007±0.001	0.007±0.002	71.7±19.8	35.4±7.1	31.4±5.3
	SAMME	0.004±0.001	0.004±0.001	0.004±0.002	2.6±0.8	3.0±0.8	2.5±1.0
	TC2	0.004±0.001	0.004±0.001	0.004±0.002	2.6±0.8	2.8±0.6	2.4±0.9
DNA	MO	0.061±0.004	0.061±0.004	0.059±0.005	97.6±4.7	492.1±12.5	990.2±19.7
	TC1	0.061±0.005	0.061±0.005	0.059±0.005	97.7±4.7	461.8±29.4	570.5±27.0
	SAMME	0.049±0.005	0.042±0.005	0.041±0.004	17.9±2.5	17.9±2.3	18.6±1.7
	TC2	0.054±0.005	0.052±0.009	0.052±0.009	20.4±4.0	36.2±3.5	48.9±11.6
Svm-guide4	MO	0.195±0.021	0.190±0.016	0.210±0.012	98.0±2.7	108.0±10.8	96.6±10.9
	TC1	0.205±0.016	0.252±0.018	0.272±0.015	52.2±14.3	32.0±3.3	32.0±2.8
	SAMME	0.160±0.018	0.174±0.016	0.158±0.015	3.0±0.0	3.0±0.0	3.0±0.0
	TC2	0.172±0.023	0.184±0.019	0.172±0.018	3.0±0.0	3.0±0.0	3.0±0.0
Svm-guide2	MO	0.209±0.032	0.214±0.032	0.221±0.035	91.9±7.6	97.9±12.1	95.5±13.0
	TC1	0.209±0.031	0.250±0.030	0.240±0.021	55.2±16.9	39.6±2.0	39.0±1.5
	SAMME	0.211±0.028	0.185±0.031	0.187±0.024	23.6±6.8	16.9±2.5	15.9±2.3
	TC2	0.215±0.019	0.203±0.024	0.189±0.024	23.5±4.2	18.9±3.7	17.3±3.5
Wine	MO	0.043±0.012	0.053±0.018	0.034±0.014	7.5±1.0	7.5±1.0	7.7±1.1
	TC1	0.057±0.036	0.057±0.029	0.043±0.019	5.3±0.8	5.3±0.8	5.9±0.5
	SAMME	0.049±0.044	0.045±0.023	0.045±0.040	3.1±0.9	3.4±0.7	2.8±0.6
	TC2	0.051±0.039	0.047±0.024	0.049±0.041	2.8±0.6	3.1±0.3	2.8±0.6
Iris	MO	0.058±0.027	0.067±0.030	0.058±0.027	34.0±14.0	34.0±14.0	44.9±13.8
	TC1	0.078±0.039	0.071±0.040	0.053±0.032	9.6±2.3	9.3±2.1	11.4±2.6
	SAMME	0.069±0.021	0.053±0.023	0.082±0.030	4.2±1.4	4.0±1.1	2.8±1.7
	TC2	0.053±0.023	0.047±0.027	0.076±0.033	3.3±0.5	3.5±0.5	2.4±0.9
Vehicle	MO	0.238±0.014	0.217±0.017	0.220±0.023	98.0±2.0	499.7±0.5	986.6±11.4
	TC1	0.236±0.014	0.223±0.021	0.220±0.026	98.0±4.1	351.3±130.5	280.7±102.6
	SAMME	0.226±0.019	0.220±0.024	0.226±0.016	24.4±2.7	17.8±1.8	29.9±5.7
	TC2	0.220±0.010	0.219±0.024	0.223±0.020	20.8±1.4	18.9±4.3	47.1±5.9
Glass	MO	0.325±0.052	0.327±0.060	0.317±0.045	94.1±5.0	102.5±10.3	102.3±14.0
	TC1	0.339±0.073	0.355±0.064	0.369±0.066	37.7±6.2	26.6±3.2	28.1±2.6
	SAMME	0.248±0.044	0.239±0.045	0.225±0.033	52.5±23.1	14.5±2.1	15.6±3.2
	TC2	0.245±0.056	0.253±0.033	0.239±0.037	24.7±6.2	9.2±3.5	10.8±3.9

process at the end of the very first step, and the strong classifier degenerates into a single decision tree. Therefore, we should add some restrictions to avoid the over-growing problem.

One of the effective methods is limiting the number of terminal nodes, which can be achieved by pruning off surplus branches after fully growing the tree. Zhu *et al.* used this kind of tree to test their algorithm SAMME. Here we make a small modification. Instead of eliminating the leaf nodes that result in slower descent of impurity, we eliminate those with smaller summation of sample weights. This method ensures the decision tree we eventually get is the one with the minimum weighted error among all the trees having the same number of nodes, in other words, the optimal weak classifier for boosting. Notice that this is not incompatible with SAMME, where the weak classifier is only required to be better than random guessing. We run SAMME and MultiBoost_{TC2} using this kind of tree in the experiment. The number of terminal nodes are chosen through a five-fold cross-validation.



Fig. 1. Examples of MNIST handwritten digit dataset

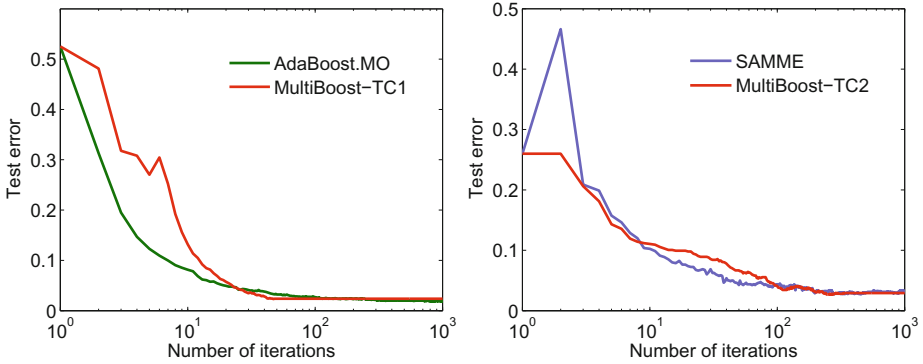


Fig. 2. Test errors on MNIST. The proposed algorithms are almost identical to their stage-wise counterparts.

The experimental results are listed in Table 1 as well. Again, the totally corrective algorithm MultiBoost_{TC2} has a faster convergence speed and comparable generalization ability with SAMME.

4.2 MNIST Handwritten Digit Dataset

In this experiment, we test the four algorithms on the MNIST dataset. MNIST contains 60,000 hand-written digit images for training and 10,000 images for testing. Some examples are shown in Figure 1. Instead of raw pixels, we run the experiment with pyramid HOG features. The filter we used to do convolution is Gaussian derivative filter with $\sigma = 2$. The number of orientation bins and dimension of features are set to be 12 and 2172, respectively.

We run each boosting learning ten times and then average the results. Each time we randomly select 10% of the data from every digit of the samples. The maximum number of training iterations is set to be 1000. The test error curves are shown in Figure 2. We can see that the results verify our earlier conclusions. The numeric results are reported in Table 2.

Table 2. Test errors and convergence steps of the four algorithms

	AdaBoost.MO	MultiBoost _{TC1}	SAMME	MultiBoost _{TC2}
test error	0.0195±0.001	0.0287±0.001	0.0340±0.001	0.0300±0.002
converg. step	52.3±2.4	45.8±2.0	7.42±0.8	6.63±0.8

5 Conclusion

In this paper, we have studied two different multi-class boosting algorithms. AdaBoost.MO decomposes a multi-class problem into multiple two-class sub-problems and then AdaBoost could be applied. SAMME directly produces an assembled classifier by using a multi-class exponential cost function. The problems they optimize are both convex, therefore, we can derive the corresponding Lagrange dual problems, and propose a column generation based framework for multi-class boosting learning. The algorithms we proposed are totally corrective and thus have faster convergence rates. Experimental results also show that our algorithms have comparable generalization capability with AdaBoost.MO and SAMME.

Acknowledgement. Work was done when Z. H. was visiting NICTA Canberra Research Laboratory and Australian National University. NICTA is funded by the Australian Government's Department of Communications, Information Technology, and the Arts and the Australian Research Council through *Backing Australia's Ability* initiative and the ICT Research Center of Excellence programs.

References

1. Freund, Y., Schapire, R.: A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *J. Comp. Syst. Sci.* 55, 119–139 (1997)
2. Viola, P., Jones, M.J.: Robust real-time face detection. *Int. J. Comp. Vis.* 57, 137–154 (2004)
3. Tieu, K., Viola, P.: Boosting image retrieval. *Int. J. Comp. Vis.* 56, 17–36 (2004)
4. Schapire, R., Singer, Y.: Improved boosting algorithms using confidence-rated predictions. *Mach. Learn.* 37, 297–336 (1999)
5. Schapire, R.: Using output codes to boost multiclass learning problems. *Mach. Learn.* 313–321 (1997)
6. Guruswami, V., Sahai, A.: Multiclass learning, boosting, and error-correcting codes. In: *Proc. Annual Conf. Learn. Theory*, pp. 145–155. ACM, New York (1999)
7. Zhu, J., Rosset, S., Zou, H., Hastie, T.: Multi-class adaboost. *Ann Arbor* 1001, 48109 (2006)
8. Shen, C., Li, H.: On the dual formulation of boosting algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1 (2010)
9. Hastie, J., Tibshirani, R.: Additive logistic regression: A statistical view of boosting. *Ann. Statist.* 28, 337–374 (2000)
10. Demiriz, A., Bennett, K., Shawe-Taylor, J.: Linear programming boosting via column generation. *Mach. Learn.* 46, 225–254 (2002)
11. Dietterich, T., Bakiri, G.: Solving Multiclass Learning Problems via Error-Correcting Output Codes. *J. Artif. Intell. Res.* 2, 263–286 (1995)
12. Sun, Y., Todorovic, S., Li, J.: Unifying multi-class adaboost algorithms with binary base learners under the margin framework. *Pattern Recogn. Lett.* 28, 631–643 (2007)

13. Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge University Press, Cambridge (2004)
14. Mosek, A.: The mosek optimization software (2007)
15. Kivinen, J., Warmuth, M.: Boosting as entropy projection. In: Proc. Annual Conf. Learn, pp. 134–144. ACM, New York (1999)

Appendix: Proof of Theorem

To derive this Lagrange dual, one needs to introduce a set of auxiliary variables $\gamma_{i,k} = -\sum_j w^{(j)} \lambda_k(y_i) \mathbf{h}_k^{(j)}(\mathbf{x}_i)$, $\forall i = 1 \dots N, k = 1 \dots K$. Then we can rewrite the primal program (4) into

$$\begin{aligned} \min_{\mathbf{w}} \quad & \sum_{i,k} \exp \gamma_{i,k} & (16) \\ \text{s.t.} \quad & \gamma_{i,k} = -\sum_{j=1}^T w^{(j)} \lambda_k(y_i) \mathbf{h}_k^{(j)}(\mathbf{x}_i), \mathbf{w} \succeq \mathbf{0}, \|\mathbf{w}\|_1 = \theta. \end{aligned}$$

By taking the constraints into the object function, we get the Lagrangian:

$$\begin{aligned} L(\mathbf{w}, \boldsymbol{\gamma}, \mathbf{u}, \mathbf{q}, r) = & \sum_{i,k} \exp \gamma_{i,k} - \sum_{i,k} u_{i,k} \left(\gamma_{i,k} + \sum_{j=1}^T w^{(j)} \lambda_k(y_i) \mathbf{h}_k^{(j)}(\mathbf{x}_i) \right) \\ & - \mathbf{q}^\top \mathbf{w} + r(\mathbf{1}^\top \mathbf{w} - \theta) \end{aligned} \quad (17)$$

with $\mathbf{q} \succeq \mathbf{0}$. The Lagrange dual function is defined as the minimum value of the Lagrangian over variables \mathbf{w} and $\boldsymbol{\gamma}$.

$$\begin{aligned} \inf_{\mathbf{w}, \boldsymbol{\gamma}} L &= \inf_{\mathbf{w}, \boldsymbol{\gamma}} \underbrace{\sum_{i,k} \exp \gamma_{i,k} - \sum_{i,k} u_{i,k} \gamma_{i,k} - r\theta}_{\text{must be } \mathbf{0}} \\ &= - \left(\sum_{i,k} u_{i,k} \lambda_k(y_i) [\mathbf{h}_k^{(1)}(\mathbf{x}_i) \cdots \mathbf{h}_k^{(T)}(\mathbf{x}_i)] + \mathbf{q}^\top - r\mathbf{1}^\top \right) \mathbf{w} \\ &= - \sum_{i,k} \overbrace{\sup_{\boldsymbol{\gamma}} (u_{i,k} \gamma_{i,k} - \exp \gamma_{i,k})}^{\text{conjugate of exponential}} - r\theta \\ &= - \sum_{i,k} (u_{i,k} \log u_{i,k} - u_{i,k}) - r\theta \end{aligned} \quad (18)$$

After eliminating \mathbf{q} we obtain the first constraint. As arguments of logarithmic functions, $\mathbf{u} \succeq 0$. The Lagrange dual problem is maximizing the dual function, and this completes the proof.

Pyramid Center-Symmetric Local Binary/Trinary Patterns for Effective Pedestrian Detection

Yongbin Zheng¹, Chunhua Shen^{2,3}, Richard Hartley^{2,3}, and Xinsheng Huang¹

¹ National University of Defense Technology, China

² NICTA, Canberra Research Laboratory, Canberra, ACT, Australia

³ Australian National University, Canberra, ACT, Australia

Abstract. Detecting pedestrians in images and videos plays a critically important role in many computer vision applications. Extraction of effective features is the key to this task. Promising features should be discriminative, robust to various variations and easy to compute. In this work, we present a novel feature, termed pyramid center-symmetric local binary/ternary patterns (pyramid CS-LBP/LTP), for pedestrian detection. The standard LBP proposed by Ojala et al. [1] mainly captures the texture information. The proposed CS-LBP feature, in contrast, captures the gradient information. Moreover, the pyramid CS-LBP/LTP is easy to implement and computationally efficient, which is desirable for real-time applications. Experiments on the INRIA pedestrian dataset show that the proposed feature outperforms the histograms of oriented gradients (HOG) feature and comparable with the start-of-the-art pyramid HOG (PHOG) feature when using the intersection kernel support vector machines (HKSVMs). We also demonstrate that the combination of our pyramid CS-LBP feature and the PHOG feature could significantly improve the detection performance—producing state-of-the-art accuracy on the INRIA pedestrian dataset.

1 Introduction

The ability to detect pedestrians in images has a major impact to applications such as video surveillance [2], smart vehicles [3, 4], robotics [5]. Changing variations in human body poses and clothing, combined with varying cluttered backgrounds and environmental conditions, make this problem far from being solved. Recently, there has been a surge of interest in pedestrian detection [6–15]. One of the leading approaches for this problem is based on sequentially applying a classifier at all the possible subwindows, which are obtained by exhaustively scanning the input image in different scales and positions. For each sliding window, certain feature sets are extracted and fed to the classifier, which is trained beforehand using a set of labeled training data of the same type of features. The classifier then determines whether the sliding window contains a pedestrian or not.

Driven by the development of object detection and classification, promising performance on pedestrian detection have been achieved by:

1. using discriminative and robust image features, such as Haar wavelets [6], region covariance [10, 12], HOG [8, 9] and PHOG [16];
2. using a combination of multiple complementary features [14];
3. including spatial information [16];
4. the choices of classifiers, such as SVMs [8, 16], AdaBoost [17].

Feature extraction is of the center importance here. Features must be robust, discriminative, compact and efficient. HOG is still considered as one of the state-of-the-art and most popular features used for pedestrian detection [8]. One of its drawbacks is the heavy computation. Maji et al. [16] introduced the PHOG feature into pedestrian detection, and their experiments showed that PHOG can yield better classification accuracy than the conventional HOG and is much computationally simpler and have smaller dimensions. However, these HOG-like features, which capture the edge or the local shape information, could perform poorly when the background is cluttered with noisy edges [14].

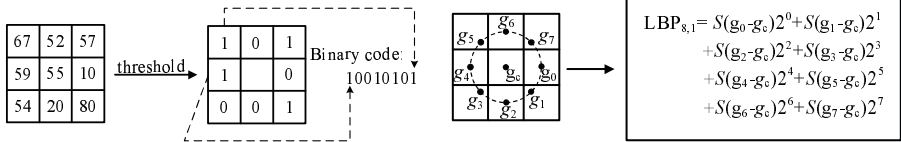
Our goal here is to develop a feature extraction method for pedestrian detection that, in comparison to the state-of-the-art, is comparable in performance but faster to compute. A conjecture is that, if both the shape and texture information are used as the features for pedestrian detection, the detection accuracy is likely to increase. The center-symmetric local binary patterns (CS-LBP) feature [18], which is a modified version of the LBP texture feature descriptor, inherits the desirable properties of both texture features and gradient based features. In addition, they are computationally cheaper and easier to implement. Furthermore, CS-LBP can be extended to center-symmetric Local Trinary Patterns (CS-LTP), which is more descriptive and less sensitive to noise in uniform image regions. In this work, we propose the pyramid CS-LBP/LTP features for pedestrian detection. Experiments on the INRIA dataset show that our new features outperform HOG and comparable with the state-of-the-art PHOG with the histogram intersection kernel SVMs (HIKSVMs) [16]. As the second contribution of this work, we show that the detection performance can be further improved significantly by combining our proposed feature with the PHOG feature.

2 Preliminaries

2.1 The LBP and LTP Features

LBP is a texture descriptor that codifies local primitives (such as curved edges, spots, flat areas) into a feature histogram. LBP and its extensions outperform existing texture descriptors both with respect to performance and to computational efficiency [1].

The standard version of the LBP feature of a pixel is formed by thresholding the 3×3 -neighborhood of each pixel with the center pixel's value. Let g_c be the center pixel graylevel and g_i ($i = 0, 1, \dots, 7$) be the graylevel of each surrounding pixel. If g_i is smaller than g_c , the binary result of the pixel is set to 0, otherwise to 1. All the results are combined to a 8-bit binary value. The decimal value of



(a) Illustration of the standard LBP operator. (b) The LBP operator of a pixel's circular neighborhoods with $r = 1, p = 8$.

Fig. 1. The LBP operator

the binary is the LBP feature. See Fig. 1 for an illustration of computing the basic LBP feature.

In order to be able to cope with textures at different scales, the original LBP has been extended to arbitrary circular neighborhoods [19] by defining the neighborhood as a set of sampling points evenly spaced on a circle centered at a pixel to be labeled. It allows any radius and number of sampling points. Bilinear interpolation is used when a sampling point does not fall in the center of a pixel. Let $LBP_{p,r}$ denote the LBP feature of a pixel's circular neighborhoods, where r is the radius of the circle and p is the number of sampling points on the circle. The $LBP_{p,r}$ can be computed as follows:

$$LBP_{p,r} = \sum_{i=0}^{p-1} S(g_i - g_c)2^i, S(x) = \begin{cases} 1 & \text{if } x \geq 0, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Here g_c is the center pixel's graylevel and $g_i (i = 0, 1, \dots, 7)$ is the graylevel of each sampling pixel on the circle. See Fig. 1 for an illustration of computing the LBP feature of a pixel's circular neighborhoods with $r = 1$ and $p = 8$.

Ojala et al. [19] proposed the concept of "uniform patterns" to reduce the number of possible LBP patterns while keeping its discrimination power. A LBP pattern is called uniform if the binary pattern contains at most two bitwise transitions from 0 to 1 or *vice versa* when the bit pattern is considered circular. For example, the bit pattern 11111111 (no transition), 00001100 (two transitions) are uniform whereas the pattern 01010000 (four transitions) is not. The uniform pattern constraint reduces the number of LBP patterns from 256 to 58 and is successfully applied to face detection in [20].

In order to make LBP less sensitive to noise, particularly in near-uniform image regions, Tan and Triggs [21] extended LBP to 3-valued codes, called local trinary patterns(LTP). If each surrounding graylevel g_i is in a zone of width $\pm t$ around the center graylevel g_c , the result value is quantized to 0. The value is quantized to +1 if g_i is above this and is quantized to -1 if g_i is below this. The $LTP_{p,r}$ can be computed as:

$$LTP_{p,r} = \sum_{i=0}^{p-1} S(g_i - g_c)3^i, S(x) = \begin{cases} 1 & \text{if } x \geq t, \\ 0 & \text{if } |x| < t, \\ -1 & \text{if } x \leq -t, \end{cases} \quad (2)$$

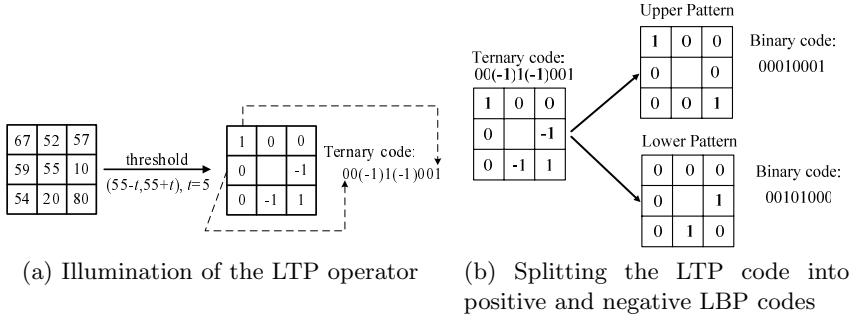


Fig. 2. The LTP operator

Here t is a user-specified threshold. Fig. 2a shows the encoding procedure of LTP. For simplicity, Tan and Triggs [21] used a coding scheme that splits each ternary pattern into its positive and negative halves as illustrated in Fig. 2b, treating these as two separate channels of LBP codings for which separate histograms are computed, combining the results only at the end of the computation.

2.2 The CS-LBP/LTP Patterns

The CS-LBP is another modified version of LBP. It is originally proposed to alleviate some drawbacks of the standard LBP. For example, the original LBP histogram could be very long and the original LBP feature is not robust on flat images. As demonstrated in Fig. 3, instead of comparing the graylevel of each pixel with the center pixel, the center-symmetric pairs of pixels are compared. The CS-LBP features can be computed by:

$$CS-LBP_{p,r,t} = \sum_{i=0}^{N/2-1} S(g_i - g_{i+(N/2)})2^i, S(x) = \begin{cases} 1 & \text{if } x \geq t, \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

Here g_i and $g_{i+N/2}$ correspond to the graylevel of center-symmetric pairs of pixels (N in total) equally spaced on a circle of radius r . Moreover, t is a small value used to threshold the graylevel difference so as to increase the robustness of the CS-LBP feature on flat image regions. From the computation of CS-LBP, we can see that the CS-LBP is closely related to the gradient operator, because like some gradient operators, it considers graylevel differences between pairs of opposite pixels in a neighborhood. In this way the CS-LBP feature takes advantage of the properties of both the LBP and gradient based features. In [18], the authors used the CS-LBP descriptor to describe the region around an interest point and their experiments show that the performance is almost equally promising as the popular SIFT descriptor. The authors also compared the computational complexity of the CS-LBP descriptor with the SIFT descriptor and it has been shown that the CS-LBP descriptor is on average 2 to 3 times

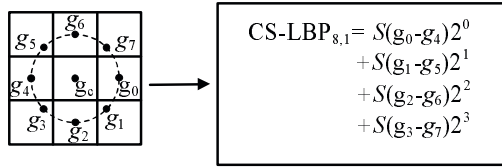


Fig. 3. The CS-LBP features for a neighborhood of 8 pixels

faster than the SIFT. That is because the CS-LBP feature needs only simple arithmetic operations while the SIFT requires time consuming inverse tangent computation when computing the gradient orientation.

Similarly as “uniform LBP patterns”, we propose “uniform CS-LBP patterns” to reduce the original CS-LBP pattern numbers. A CS-LBP pattern is called uniform if the binary pattern contains at most one bitwise transition from 0 to 1 or vice versa. For example, patterns 0000 (no transition) and 0111 (one transition) are uniform whereas patterns 0010 (two transitions) and 1010 (three transitions) are not. We computed the CS-LBP patterns of 741 images in the INRIA dataset (288 images containing pedestrians and 453 images without pedestrians) and found that 87.82% of the patterns are uniform, shown in Table 1.

The CS-LTP patterns and the uniform CS-LTP patterns can be developed similarly as the CS-LBP and the uniform CS-LBP.

Table 1. The distribution of the CS-LBP patterns (uniform and non-uniform) on the INRIA pedestrian dataset

Uniform pattern	0000	0001	0011	0111	1000	1100	1110	1111	Total
Percent. (%)	8.93	11.80	8.72	10.22	8.31	9.27	10.99	19.57	87.82
Non-uniform pattern	0010	0100	0101	0110	1001	1010	1011	1101	Total
Percent. (%)	1.24	1.14	1.52	1.28	1.86	1.31	1.73	2.11	12.18

2.3 The Pyramid CS-LBP/LTP Features and Pyramid Uniform CS-LBP/LTP Features

Motivated by the image pyramid representation in [22] and the HOG feature [8], Bosch et al. [23] proposed the PHOG descriptor, which consists of a pyramid of histograms of orientation gradients, to represent an image by its local shape and the spatial layout of the shape. Experiments showed that PHOG feature together with the histogram intersection kernel can bring significant performance to object classification and recognition. Maji et al. [16] introduced the PHOG feature into pedestrian detection and achieved the current state-of-the-art on pedestrian detection. In this study, we propose the pyramid CS-LBP/LTP features. Because the LTP patterns can be divided into two LBP patterns, we only

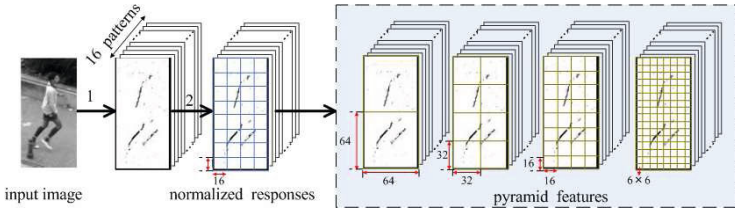


Fig. 4. The first three steps of computing the pyramid CS-LBP feature. (1)Edge energy responses corresponding to each CS-LBP pattern of the input 128×64 image are computed. (2)The responses are L_1 normalized over all layers in each non overlapping 16×16 cells independently so that the normalized gradient values in each cell sum to unity. (3)The features at each level is extracted by concatenating the histograms, which are constructed by summing up the normalized response within each cell at the level. The cell size at level 1,2,3 and 4 are $64 \times 64, 32 \times 32, 16 \times 16$ and 6×6 respectively.

illustrate the computation of the pyramid CS-LBP features. Our features of a 64×128 detection window are computed as follows (Fig. 4 shows the first three steps of computing the features):

1) We compute the CS-LBP value and the norm of each pixel of the input grayscale image(detection window). The LBP value is computed as Eq. 3 with $t = 0.015$ and the norm of the pixel located at (x, y) is computed as: $norm(x, y) = \sqrt{G_x^2(x, y) + G_y^2(x, y)}$, where $G_x(x, y)$ and $G_y(x, y)$ are the horizontal gradient and vertical gradient of the pixel. Then we obtain 16 layers of norm images corresponding to each CS-LBP pattern. We call them edge energy responses of the input image. Fig. 5 shows the 8 layers of edge energy responses of the example image corresponding to each uniform CS-LBP pattern respectively. The 8 layers of edge energy responses corresponding to non-uniform CS-LBP patterns are not plotted due to space limit.

2) Each layer of the response image is L_1 normalized in non overlapping cells of fixed size $y_n \times x_n$ ($y_n = 16, x_n = 16$) so that the normalized gradient values in each cell sum to unity.

3) At each level $l \in \{1, 2, \dots, L\}$, the response image is divided into non overlapping cells of size $y_l \times x_l$, and a histogram with 16 bins is constructed by summing up normalized response within the cell. In our case, $L = 4, y_1 = x_1 = 64, y_2 = x_2 = 32, y_3 = x_3 = 16, y_4 = x_4 = 6$. So we obtain 2, 8, 32, and 210 histograms at level $l = 1, 2, 3$ and 4 respectively.

4) The histograms of each level is normalized to sum to unity. This normalization ensures that the edge or texture rich images are not weighted more strongly than others.

5)The features at a level l are weighted by a factor w_l ($w_1 = 1, w_2 = 2, w_3 = 4, w_4 = 9$), and the features at all the levels are concatenated to form a vector of dimension 4, 032, which is called pyramid CS-LBP features.

The process of computing pyramid uniform CS-LBP features is almost same as pyramid CS-LBP. The only difference lies in the first step. In the first step,

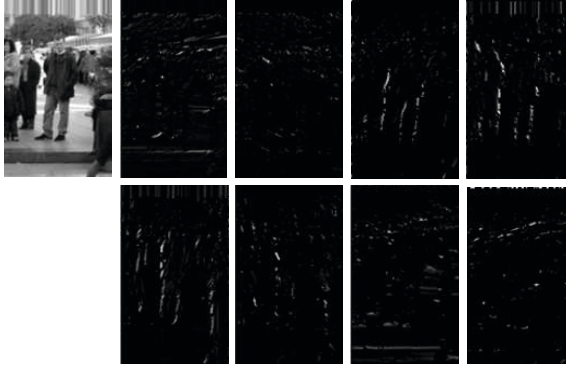


Fig. 5. Edge energy responses of an example image. The first image is the input image and the rests are its 8 layers of edge energy responses corresponding to the 8 uniform CS-LBP pattern. The 8 layers of edge energy responses corresponding to the 8 non-uniform patterns are not shown due to space limit.

the edge energy responses corresponding to the 8 different uniform patterns are count into 8 different layers and the edge energy response corresponding to all the 8 non-uniform patterns are count into one layer. So we obtain 9 layers of edge energy responses of the input image.

3 Pedestrian Detection Based on Pyramid CS-LBP/LTP Features

We use the sliding window approach. The first major component of our approach is feature extraction. We perform the graylevel normalization of the input image to deduce the illumination variance. After the normalization is performed, all the input image have the graylevel ranged from 0 to 1. Then the detection window slides on the input images in all positions and scales, with a fixed step size 8×8 and a fixed scale factor 1.0905. We follow the steps in Sec. 2.3 to compute the pyramid CS-LBP/LTP features of each 64×128 detection window.

The second major component of our approach is the classifier. We use histogram intersection kernel SVMs (HIKSVMs) [16] as the classifier. The histogram intersection kernel, $k_{HI}(h_a, h_b) = \sum_{i=1}^n \min(h_a(i), h_b(i))$ is often used as a measurement of similarity between histogram h_a and h_b and it can be used as a kernel for classification using SVMs. Compared to linear SVMs, histogram intersection kernel involves great computational expense. Maji et al. [16] approximated the histogram intersection kernel for faster execution. Their experiments showed that the approximate HIKSVMs consistently outperforms linear SVMs at a modest increase in running time.

The third major component of our approach is the merging of the multiple overlapping detections using non maximal suppression(NMS). After merging, detections with bounding boxes and confidence scores are obtained.

4 Experiments

4.1 Experiment Setup

Datasets. We perform the experiments on INRIA pedestrian dataset [8], which is one of the most popular publicly available datasets. The dataset consists of a training set and a test set. The training set contains 1,208 images of size 96×160 pixels (a margin of 16 pixels around each side) of human samples (2,416 mirrored samples) and 1,218 pedestrian-free images. The test set contains 288 images with human samples and 453 human free images. The human samples are cropped from a varied set of personal photos and vary in pose, clothing, illumination, background and partial occlusions, what make the dataset is very challenge.

Methodology. *Per-window* performance is accepted as the methodology for evaluating pedestrian detectors by most researchers. But this evaluating methodology is flawed. As pointed out in [13], *per-window* performance can fail to predicate *per-image* performance. There may be at least two reasons: first, *per-window* evaluation does not measure errors caused by detections at incorrect scales or positions or arising from false detections on body parts, nor does it take into account the effect of non maximal suppression. Second, the *per-window* scheme uses cropped positives and uncropped negatives for training and testing; classifiers may exploit window boundary effects as discriminative features leading to good *per-window* but poor *per-image* performance. In this paper, we use *per-image* performance, plotting detection rate versus false positives per-image(FPPI).

We select the 2,416 mirrored human samples from the training set as positive training examples. A fixed set of 12,180 patches sampled randomly from 1,218 pedestrian-free training images as initial negative set. As in [8], a preliminary HIKSVMS detector is trained and the 1,218 negative training images are searched exhaustively for false positives(‘hard examples’). The final classifier is then trained using the augmented set(initial 12,180 + hard examples). The SVMs tool we used is the fast intersection kernel SVMs proposed by Maji et al. [16].

We detect pedestrians on each test images (both positive and negative) in all positions and scale with a step size 8×8 and a scale factor 1.0905. Multiscale and nearby detections are merged using NMS and a list of detected bounding boxes are given out. Evaluation on the list of detected bounding box is done using the PASCAL criterion which counts a detection to be correct if the overlap of the detected bounding box and ground truth bounding box is greater than 0.5.

4.2 Performance of the Pyramid CS-LBP/LTP Feature Based Detector

In this section, we study the performance of our approach by comparing with the state of art PHOG feature based approach. We obtain the PHOG based detector from its author, and all the parameters of the PHOG(such as the L_1 normalization cell size, the level number and cell size in each level) are same

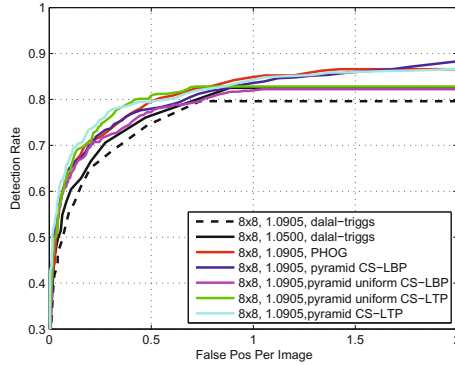


Fig. 6. Detection rate versus false positive per-image (FPPI) curves for detectors based on the pyramid CS-LBP/LTP features using HIKSVMs classifier, the pyramid uniform CS-LBP/LTP features using HIKSVMs classifier, the PHOG feature using HIKSVMs classifier and the HOG feature using linear SVMs classifier. 8×8 is the step size and 1.0905 is the scale factor of the sliding detection window.

as our features. The results are shown in Fig. 6. The performance of pyramid CS-LTP based detector performs best, with detection rate over 80% at 0.5 FPPI. Then followed by the pyramid uniform CS-LTP based detector, which is slightly better than the PHOG based detector. The pyramid CS-LBP based detector performs almost as good as the PHOG. Though the pyramid uniform CS-LBP based detector performs slightly worse than PHOG based detector, it outperforms the HOG features with linear SVMs based detector proposed by Dalal and Triggs [8].

4.3 Detection Results with Features Combined with Pyramid CS-LBP and PHOG

In this experiment, our main aim is to find out whether the combination of our feature with PHOG feature can achieve better detection result or not. We use the following simplest method to combine the pyramid uniform CS-LBP feature with the PHOG feature [24]:

$$K_c(v_1, v_2) = 0.5K_1(v_1) + 0.5K_2(v_2) \quad (4)$$

where K_1 and K_2 are the HIKSVMs classifiers pretrained using the pyramid uniform CS-LBP feature and the PHOG feature respectively, v_1 and v_2 are the pyramid uniform CS-LBP feature and the PHOG feature of a detection window respectively.

Detection performance are shown In Fig. 7. The detection rate versus FPPI curves show that the feature combination can significantly improve the detection performance. Compared to the PHOG, the detection rate raises about 6% at 0.25

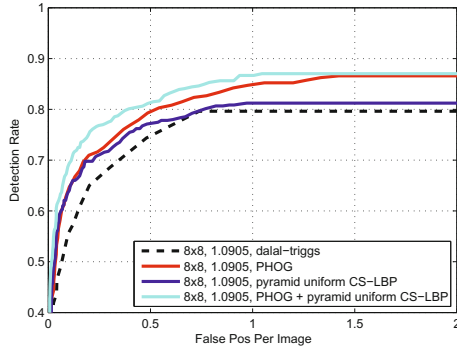


Fig. 7. Detection rate versus false positive per-image(FPPI) curves for detectors(using HIKSVMs classifier) based on the PHOG features, the uniform CS-LBP feature and the augmented features combined by the HOG and the pyramid uniform CS-LBP. The augmented feature can improve the detection accuracy significantly. 8×8 is the step size and 1.0905 is the scale factor of the sliding detection window.



Fig. 8. Some examples of detections on test images for the detectors using PHOG, pyramid uniform CS-LBP and augmented features (combined with HOG and pyramid uniform CS-LBP). First row: detected by the PHOG based detector. Second row: detected by the pyramid uniform CS-LBP based detector. Third row: detected by the PHOG+pyramid uniform CS-LBP based detector.

FPPI and raises about 1.5% at 0.5 to 1 FPPI. Fig. 8 shows pedestrian detection on some example test images. The three rows show the bounding boxes detected by PHOG based detector, the pyramid uniform CS-LBP based detector and the PHOG + pyramid uniform CS-LBP based detector, respectively.

5 Conclusions

Experimental results on the INRIA dataset show that the pyramid CS-LTP features using the HIKSVMs classifier outperform the PHOG, and the pyramid CS-LBP features perform as well as the HOG feature. We have also show that combining the pyramid CS-LBP with PHOG produces a significantly better detection performance on the INRIA dataset.

There are many directions for further research. To make the conclusion more convincing, the performance of the pyramid CS-LBP/LTP features based pedestrian detector needs to be further evaluated on other dataset, e.g., the Caltech Pedestrian Dataset [13]. Another further study is to compare the computational complexity of the pyramid CS-LBP/LTP features with PHOG both theoretically and experimentally. Thirdly, it is worthy studying how to combine our features with PHOG or other features more efficiently. We are also interested in implement the new feature in a boosting framework.

Acknowledgement. Work was done when Y. Zheng was visiting NICTA Canberra Research Laboratory and Australian National University. NICTA is funded by the Australian Government's Department of Communications, Information Technology, and the Arts and the Australian Research Council through *Backing Australia's Ability* initiative and the ICT Research Center of Excellence programs.

References

1. Ojala, T., Pietikainen, M., Harwood, D.: A comparative study of texture measures with classification based on featured distribution. *Pattern Recogn.* 29, 51–59 (1996)
2. Haritaoglu, I., Harwood, D., Davis, L.S.: W^4 : real-time surveillance of people and their activities. *IEEE Trans. Pattern Anal. Mach. Intell.* 22, 809–830 (2000)
3. Gavrila, D.M., Munder, S.: Multi-cue pedestrian detection and tracking from a moving vehicle. *Int. J. Comp. Vis.* 73, 41–59 (2007)
4. abd J. Giebel, D.M.G., Munder, S.: Vision-based pedestrian detection: the protector system. In: *Proc. IEEE Int. Conf. Intell. Vehic. Symposium*, Parma, Italy, pp. 13–18 (2004)
5. Nakada, T., Kagami, S., Mizoguchi, H.: Pedestrian detection using 3d optical flow sequences for a mobile robot. In: *Proc. IEEE Conf. Sens*, Lecce, Italy, pp. 776–779 (2008)
6. Viola, P., Jones, M.J., Snow, D.: Detecting pedestrian using patterns of motion and appearance. In: *Proc. IEEE Int. Conf. Comp. Vis.*, Nice, France, vol. 2, pp. 734–741 (2003)
7. Mikolajczyk, K., Schmid, C., Zisserman, A.: Human detection based on a probabilistic assembly of robust part detectors. In: Pajdla, T., Matas, J(G.) (eds.) *ECCV 2004*. LNCS, vol. 3021, pp. 69–82. Springer, Heidelberg (2004)
8. Dalal, N., Triggs, B.: Histogram of oriented gradients for human detection. In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, San Diego, USA, vol. 1, pp. 886–893 (2005)
9. Dalal, N.: Finding people in images and videos. PhD thesis, Institut National Polytechnique de Grenoble (2006)

10. Tuzel, O., Porikli, F., Meer, P.: Human detection via classification on riemannian manifolds. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn., Minneapolis, Minnesota, USA, pp. 1–8 (2007)
11. Munder, S., Gavrila, D.M.: An experimental study on pedestrian classification. *IEEE Trans. Pattern Anal. Mach. Intell.* 28, 1863–1868 (2006)
12. Paisitkriangkrai, S., Shen, C., Zhang, J.: Fast pedestrian detection using a cascade of boosted covariance features. *IEEE Trans. Circuits Syst. Video Technol.* 18, 1140–1151 (2008)
13. Dollár, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: A benchmark. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn., pp. 304–311 (2009)
14. Wang, X., Han, T., Yan, S.: An HoG–LBP human detector with partial occlusion handling. In: Proc. IEEE Int. Conf. Comp. Vis., Kyoto, Japan, pp. 32–39 (2009)
15. Enzweiler, M., Gavrila, D.M.: Monocular pedestrian detection: Survey and experiments. *IEEE Trans. Pattern Anal. Mach. Intell.* 31, 2179–2195 (2009)
16. Maji, S., Berg, A., Malik, J.: Classification using intersection kernel support vector machines is efficient. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn., Anchorage, Alaska, USA, pp. 1–8 (2008)
17. Wojek, C., Walk, S., Schiele, B.: Multi-cue onboard pedestrian detection. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn., Miami, Florida, USA, pp. 794–801 (2009)
18. Heikkilä, M., Schmid, C.: Description of interest regions with local binary patterns. *Pattern Recogn.* 42, 425–436 (2009)
19. Ojala, T., Pietikainen, M., Maenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* 24, 971–987 (2002)
20. Ahonen, T., Hadid, A., Pietikainen, M.: Face detection with local binary patterns: application to face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 28, 2037–2041 (2006)
21. Tan, X., Triggs, B.: Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Trans. Image Process.* 19, 1635–1650 (2010)
22. Bosch, A., Zisserman, A., Muñoz, X.: Scene classification via PLSA. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) *ECCV 2006*. LNCS, vol. 3954, pp. 517–530. Springer, Heidelberg (2006)
23. Bosch, A., Zisserman, A., Muñoz, X.: Representing shape with a spatial pyramid kernel. In: Proc. ACM. Int. Conf. Image & Video Retrieval, pp. 401–408 (2007)
24. Gehler, P., Nowozin, S.: On feature combination for multiclass object classification. In: Proc. IEEE Int. Conf. Comp. Vis., pp. 221–228 (2009)

Reducing Ambiguity in Object Recognition Using Relational Information

Kuk-Jin Yoon and Min-Gil Shin

Computer Vision Laboratory
School of Information and Communication, GIST, Republic of Korea
kjyoon@gist.ac.kr

Abstract. Local feature-based object recognition methods recognize learned objects by unordered local feature matching followed by verification. However, the matching between unordered feature sets might be ambiguous as the number of objects increases, because multiple similar features can be observed in different objects. In this context, we present a new method for textured object recognition based on relational information between local features. To efficiently reduce ambiguity, we represent objects using the Attributed Relational Graph. Robust object recognition is achieved by the inexact graph matching. Here, we propose a new method for building graphs and define robust attributes for nodes and edges of the graph, which are the most important factors in the graph-based object representation, and also propose a cost function for graph matching. Dependent on the proposed attributes, the proposed framework can be applied to both single-image-based and stereo-image-based object recognition.

1 Introduction

One main issue in object recognition is how to cope with appearance variations caused by photometric and geometric changes. In this point of view, a local invariant feature-based approach is one faithful solution for robust object recognition. In this approach, each object is represented by the set of unordered local features which are invariant to photometric and geometric variations. This approach is generally composed of several steps.

The first step is visual part detection. Lindeberg [1] proposed a method on blob-like image structure detection in scale space. Shokoufandeh et al. [2] extended this feature to wavelet domain. Schmid et al. [3] compared various interest point detectors and concluded that the scale-reflected Harris corner detector is the most robust to image variations. Mikolajczyk and Schmid [4] also compared visual part extractors and found that the Harris-Laplacian-based part detector is suitable for most applications. The next step is to generate proper descriptors of extracted features for matching. Recently several visual descriptors have been proposed [5-9]. Most approaches try to encode local visual information such as spatial orientation or edgeness. Based on these local visual features and their descriptors, several object recognition methods, such as the probabilistic

voting method [10] and constellation model-based approach [11, 12] have been introduced as well.

Here, it is noteworthy that most approaches are highly dependent on simple descriptor matching. Learned objects are generally represented by using unordered local feature sets and recognition is achieved by the matching local features in unordered feature sets. However, the matching between unordered feature sets might be ambiguous and erroneous as the number of objects increases, because multiple similar features can be observed in different objects. Therefore, it is very important to reduce the inherent ambiguity owing to ambiguous local features in local feature-based object recognition.

To reduce the inherent ambiguity, it is very helpful to use not only individual features but also groups of multiple features and/or their relational information together. This is motivated by the observation that, although similar local features can be extracted from multiple objects, it is quite rare that a set of local features with specific relation is extracted from multiple objects.

In early works such as generalized Hough transform [13] and geometric hashing [14], the object was represented as a point cloud because locations of features (such as corner and edge) were known. [13] used distance and orientation between a reference point and edge points in order to represent arbitrary object shape, and in [14], arbitrary small points are selected as a basis and coordinates of transformed points according to basis are used. However these method require huge memory and computation and are sensitive to noise.

An object can be represented as the bag of features [7, 15, 16]. Each object is recognized based on the feature descriptor matching. However, the result can be ambiguous due to ambiguous local features. Kim et al. [6] presented a new recognition method based on the Gestalt's grouping law to utilize high-level context information between individual local features. Similarly, part-based models [17, 18] also have been proposed to represent an object with the spatial information (mainly defined in the image domain). A similar configuration of parts is found by solving an optimization problem related to the matching model. However, since these works use image coordinates to encode the relational information, they are weak to heavy appearance variations.

On the other hand, there are some works trying to use graph for feature matching and/or object recognition [19–24], since graph is also an appropriate data structure to impose relational information between features. However, most works focus on graph matching strategy. [20] and [21] find feature correspondence via graph matching. In [22], authors recognized and tracked limited object in video sequence under limited environments. Recently, [25] adopts a hash table to find potential corresponding points between objects.

In this context, we present a new method for textured object recognition based on a graph. To efficiently reduce the ambiguity owing to the similar local features, we represent objects using the Attributed Relational Graph(ARG). Accurate and robust object recognition is achieved by using graph-based object representation, which contains both local features and the relative relationship between local features, and by using the inexact graph matching. Here, we

propose a new method for building graphs and define robust attributes for nodes and edges of the graph, which are the most important factors in the graph-based object representation, and also propose a cost function for graph matching. Dependent on the proposed attributes, the proposed framework can be applied to both single-image-based and stereo-image-based object recognition. The proposed method works for general 3D objects and can deal with partially occluded objects in the cluttered scene. In addition, it shows robust performance against scale and view-point changes.

2 Overall Structure of the Proposed Method

There are two key issues in local feature-based object recognition methods: (1) local feature selection and (2) recognition strategy. The first issue is related to extraction of proper local features and the second issue is related to the recognition scheme using extracted local features. While both issues are equally important, we focus on the second issue, especially on the object representation and their matching for candidate selection. Here, although we adopt SIFT [7] for our experiments, it is possible to adopt any other local features in our framework.

The overall structure of the proposed framework is shown in Fig. 1. The proposed object recognition method consists of three stages. The first stage is the extraction of local features from an input image. In this stage, local features and their descriptors are obtained and then fed into the object representation stage. As mentioned, we represent objects with graphs to reduce the ambiguity in recognition. The object ARG is generated based on a local feature descriptor and relation between any two neighbor local features. The third stage is the recognition stage using extracted local features and the object representation. In this stage, good candidates (i.e., recognition hypothesis) are selected first by using inexact graph matching and then the selected candidates are examined by the verification method (hypothesis testing).

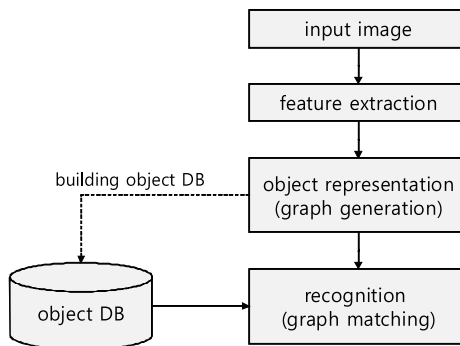


Fig. 1. Overall structure of the proposed framework

3 Object Recognition with ARG

3.1 ARG-Based Object Representation

The proper object representation is the key to reducing the ambiguity in local feature-based object recognition, which is owing to ambiguous appearance of local features from different objects. In previous works as the bag of words method [26], an object is commonly represented by the set of unordered individual features as

$$O \leftarrow F = \{f(i) | 1 \leq i \leq n\}, \quad (1)$$

where n is the number of local features belonging to the object O . However, we cannot avoid the inherent ambiguity when using this naive representation. Therefore, we need more contextual object representation.

A slightly more complex but very efficient approach is to represent each object by using pair-wise local features as well as individual local features as

$$O \leftarrow F \cup P, \quad (2)$$

where

$$P = \{(f(i), f(j), \nu(i, j)) | f(i) \in F, f(j) \in F, i < j\}. \quad (3)$$

Here, $\nu(i, j)$ is the relational information between two features $f(i)$ and $f(j)$. By combining two local features and their relational information together, we can greatly reduce the ambiguity of individual local features. This is motivated by the observation that, although similar local features can be extracted from multiple objects, it is quite rare that a pair of local features $(f(i), f(j))$ with $\nu(i, j)$ is extracted from multiple objects.

More generally, it is also possible to build a graph to represent each object, which is very well-suited and powerful data structure and widely used in structural and semantic pattern recognition as in [24]. In particular, an ARG is a graph in which attribute vectors are assigned to vertices and to edges. Such vectors are responsible for adding relevant problem information to the graph data structure, since they hold symbolic properties and features related to the nodes and edges they are assigned to. When adopting an undirected ARG for object representation, it can be expressed as

$$O \leftarrow G := (V, E, \mu, \nu). \quad (4)$$

Here, V is a set of vertices and E is a set of edges that represent the relation between vertices. In addition, μ is the attribute of vertex and ν is the attribute of edge (or weight of edge). μ and ν can be vectors or scalar values depending on the definition. In fact, edges and their attributes play an important role in graph matching, and therefore, the criterion of defining E and ν is very significant. The issue in defining E is determining the number of edges per a vertex. If a vertex has edges with all other vertices, graph matching takes huge time and the method becomes impractical. For that reason, in the proposed method, a vertex has a small number of edges with K -nearest-neighbor vertices. On the other hand, the issue in defining ν is finding invariant property between any two vertices robust against photometric and geometric changes.

3.2 Inexact Graph Matching

In the learning stage, we construct $G := (V, E)$ for all object to be recognized. In this paper, we assume that we have M objects to recognize, and we denote $G_m := (V_m, E_m)$ for object O_m , $1 \leq m \leq M$. The local feature from O_m is denoted as f_m .¹ Once we get an input image, we extract local features and construct $G_s := (V_s, E_s)$ using extracted local features. Then, recognition hypothesis is generated by comparing $G_s := (V_s, E_s)$ with those in the database.

Since objects and in input image are represented by graph, the problem of selecting good candidates is changed into the problem of graph matching. Here, the matching between two graphs G_s and G_m can be represented by a matching matrix $Q(s, m)$. For two graphs G_s and G_m , we find $Q(s, m)$ that minimizes the objective function defined as in [19] like

$$\Omega(Q) = -\frac{1}{2} \sum_{a=1}^{n'_s} \sum_{i=1}^{n'_m} \sum_{b=1}^{n'_s} \sum_{j=1}^{n'_m} Q_{ai} Q_{bj} c_{aibj} + \alpha \sum_{a=1}^{n'_s} \sum_{i=1}^{n'_m} Q_{ai} c_{ai} \quad (5)$$

subject to

$$\forall a, \sum_{i=1}^{n'_m} Q_{ai} \leq 1, \quad \forall i, \sum_{a=1}^{n'_s} Q_{ai} \leq 1, \quad \text{and} \quad \forall a, \forall i Q_{ai} \in \{0, 1\}. \quad (6)$$

Here, n_s and n_m are the number of vertices of G_s and G_m , respectively. The matching matrix Q is the $n_s \times n_m$ matrix indicating which vertices in the two graphs match as

$$Q_{ai} = \begin{cases} 1 & \text{if vertex } a \in G_s \text{ corresponds to vertex } i \in G_m \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

c_{aibj} represents the similarity of two edges $\{a, b\}$ and $\{i, j\}$. We define it as

$$c_{aibj} = 1 - \frac{|\nu_{ab} - \nu_{ij}|}{\nu_{ab} + \nu_{ij}} \quad (8)$$

if $\nu_{ab} \neq 0$ and $\nu_{ij} \neq 0$. Otherwise, $c_{aibj} = 0$. On the other hand, c_{ai} represents the dissimilarity of two vertices a and i . We also define it as

$$c_{ai} = 1 - \mu_a \cdot \mu_i \quad (9)$$

where μ_a and μ_i are the attributes of vertex a and i , respectively. α in Eq. (8) controls the weight of the vertex matching cost. Here, the range of c_{aibj} and c_{ai} is from 0 to 1. Because each vertex has edges with K -nearest-neighbor vertices, the range of sum of c_{ai} of each vertex is from 0 to 1 and the range of sum of c_{aibj} of each vertex is from 0 to K^2 . Therefore, the number of edges per each vertex

¹ Subscript m denotes the object index while subscript s denotes the input.

seriously affects the value of objective function [5]. Therefore, α is dependent on the number of edges per each vertex.

Since this is the inexact graph matching problem, we can deal with occlusion or any other variation in graphs. We compute the matching matrix between graphs by using a graduated assignment algorithm in [19]. Once we compute the all matching matrices between G_s and all G_m s, we choose objects that have the matching matrices satisfying $\sum_{a=1}^{n'_s} \sum_{i=1}^{n'_m} Q_{ai} > T_g$ as recognition candidates.

The proposed method can be applied to both single-image-based and stereo-image-based object recognition depending on the definition of ν .

4 Graph-Based Approach in a Single Image

In a single image, we have to make graph based on information offered by a local feature extraction method. After generating object graphs and input graph, we conduct inexact matching to get good candidates. Since we have 2D image coordinates of extracted local features, verification can be performed by using 2D homography or fundamental matrix.

As shown in Fig. 1, the first phase is to extract proper local features and the second phase is to generate ARG, $G = (V, E, \mu, \nu)$. In our ARG, each vertex $v \in V$ corresponds to an extracted individual local SIFT feature [7] and μ is the local feature's descriptor vector of which the norm is 1. We define ν by using the difference between two vertices' orientations, which is invariant to geometric variations [2]. Here, it is also possible to define ν in a different way when adapting different local features.

Each vertex is connected with only K -nearest-neighbor vertices to reduce computation time in graph matching [3]. Here, K -nearest-neighbor vertices are computed by comparing the orientation difference, because features having similar orientations are more likely to have similar orientations under geometric changes too. However, when an edge is created just by the order of orientation difference, an edge can be created between uncorrelated vertices (for instance, one from the object of interest and one from the background). In order to solve this problem, we select K -nearest-neighbor vertices within a specified radius (in an image). Therefore, for two nodes u and v in V , if v is the K -nearest-neighbor of u in a specified radius, then ν of the edge $\{u, v\}$, ν_{uv} is defined as the orientation difference between two vertices. Otherwise $\nu_{uv} = 0$.

The next stage is performing inexact graph matching to get good candidates. It is accomplished by the method described in Sec. 3.2. The selected candidates are then verified in the last phase. Because we have 2D image coordinates of matched local features in a single image, we conduct verification using 2D homography or fundamental matrix, which can be computed with the RANSAC technique.

² The orientation of each local feature in an image can vary under geometric variations. However, the relative difference between two features' orientations is relatively invariant under geometric variations.

³ We set $K = 10$ for experiments.

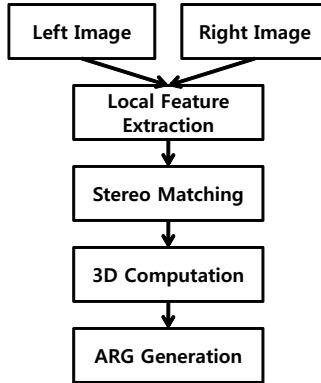


Fig. 2. ARG generation in a stereo image

5 Graph-Based Approach in a Stereo Image

In a stereo image, an object can be more robustly recognized than a single image. We can secure wider field of view (FOV) thanks to the two cameras. In addition, we can achieve more robust object recognition under the viewpoint change and occlusion, because we have two input images of the same object from two different viewpoints. Lastly, we can obtain 3D coordinates of local features that make it possible to easily handle 3D objects.

In this case, although we are able to make graph as in a single image case, we use additional information such as 3D coordinates of local features. The overall process of object recognition is similar to that in a single image case. However, additional stages such as stereo matching and computation of local features' 3D coordinates are required as in Fig. 2.

Local features are extracted first using [7] and the extracted features from a left image and a right image are then matched across images. Since the stereo images are rectified and there are little changes in scale, orientation, and illumination between stereo images, we can achieve very reliable stereo matching in real time. Here, we apply the unique-minimum check and the left-right consistency check to remove ambiguous matches, which are commonly used for stereo matching. After finding correspondences, we compute the 3D coordinates of matched local features with respect to the stereo camera.

The next phase is to generate ARG, $G = (V, E, \mu, \nu)$. In this case, each vertex $v \in V$ corresponds to a stereo matched local feature (having computed 3D coordinates) and μ is the local feature's descriptor vector of which the norm is 1. Here, we define ν by using the 3D Euclidean distance between stereo matched local features, which is invariant to any geometric variations (when assuming rigid objects). Therefore, for two nodes u and v in V , if v is the K -nearest-neighbor of u that satisfies $D(u, v) \geq T$, then ν of the edge $\{u, v\}$, ν_{uv} is defined as $\nu_{uv} = w_{uv}D(u, v)$. Otherwise $\nu_{uv} = 0$. Here, $D(u, v)$ is the 3D Euclidean



Fig. 3. Some stereo training images

distance between node u and node v , and T is pre-defined threshold. w_{uv} is weight of $D(u, v)$ and defined as

$$w_{uv} = \frac{1}{s_u + s_v} + \epsilon, \quad (10)$$

where s_u and s_v are the scales of nodes u and v inferred from stereo matched local features. We assign small weights to nodes having large scales, because the accuracy of their 3D coordinates can be poor. 0.5 is used for ϵ in our experiments.

The next stage is the inexact graph matching to get good candidates. It is solved by the method in Sec. 3.2.

The verification is performed similarly in the single image case. The difference is that we use 3D rigid transformation instead of the 2D homography or fundamental matrix, because we already have 3D coordinates of stereo matched local features. The 3D rigid transformation can be described with one rotation matrix, R , and one translation vector, t , and they are computed with the RANSAC technique as well.

6 Experiments and Performance Evaluation

6.1 Image Database

We built the image database of 3D objects in order to evaluate the proposed method and to compare the performance. It contains 4,500 rectified stereo images of 100 different 3D textured objects (45 stereo images for each object), which were captured under controlled darkroom environments and the cluttered office environments⁴. For each object, 36 images were captured under the controlled darkroom environments and 9 images were captured under office environments. More specifically, for each object, images were captured while changing the illumination intensity from 110 lux to 270 lux in nine steps. In the same manner, we captured images under scale (about 2 times \sim 0.5 times) and yaw ($-30^\circ \sim 30^\circ$)

⁴ The database and its detailed description can be found at <http://cvl.gist.ac.kr>

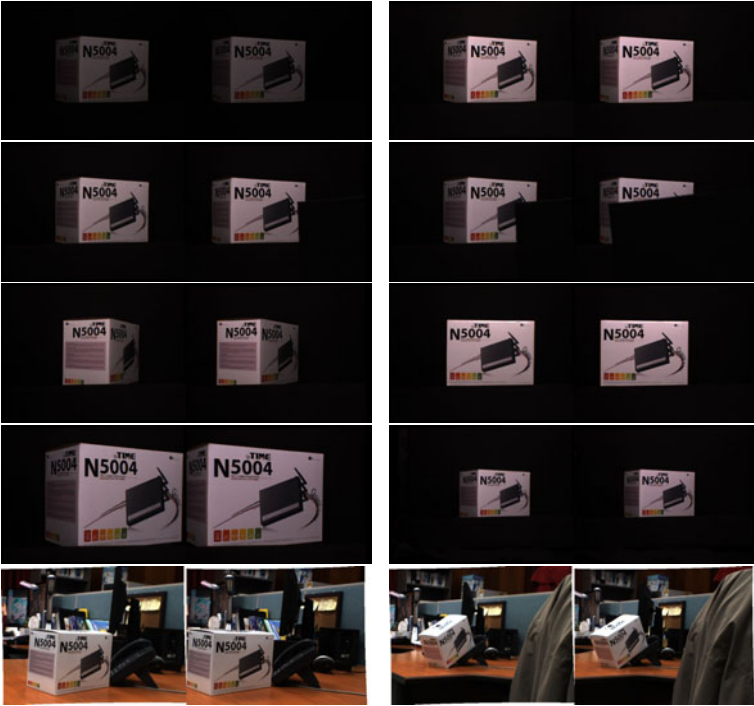


Fig. 4. Test images captured under different conditions

changes in nine steps, respectively. In addition, we captured images for each object while controlling the total amount of occlusion (5% ~ 45%). Some training and test images are shown in Fig. 3 and Fig. 4, respectively.

6.2 Experiments and Analysis

We first applied the proposed method to a single image. Although we have the object database with stereo images, we used single images (i.e., left images) instead of stereo images in this experiment. For learning objects, although it is feasible to use multiple images of the same object obtained from different conditions for learning, we used just one frontal single image of each object for learning in our experiments, to clearly see the performance enhancement by proposed method. In addition, as mentioned earlier, we adopt the SIFT [7] for our experiments.

In the single descriptor matching method, candidate objects are selected by using a simple voting technique based on the feature descriptor matching. We find correspondences of local features of an input image through the criterion defined as

$$\frac{d(f(i), d(f(NN_{1st}))}{d(f(i), d(f(NN_{2nd}))} < 0.49, \quad (11)$$

where $f(i)$ is a local feature of an input image, $f(NN_{1st})$ is the first nearest neighbor feature of $f(i)$, $f(NN_{2nd})$ is the second nearest neighbor feature of $f(i)$, and $d(f(i), f(j))$ is the descriptor difference between two local features. The candidates are rejected if they fail to pass the verification stage.

In the single graph-based method (graph-based method for a single image), we constructed graphs, matched graphs, and verified the results as in Sec. 4. Verification criterion is the same as in the descriptor matching method.

We also applied the proposed method to stereo images. In this case, we used one frontal stereo image of each object for learning. In this case, we made graphs, matched graphs, and verified the results as in Sec. 5. Verification criterion is the same as in other methods except the inlier criterion — we used 3D Euclidean distance to decide the inliers.

The performance for a single descriptor matching method, a single graph-based method, and a stereo graph-based method is given in Table 1, Table 2, Fig. 5, and Fig. 6. For each case, we adjusted parameters to obtain the best correct ratio. Correct, error, false positive, and false negative ratios are defined as

$$\text{correct ratio}(\%) = \frac{\# \text{ of correct object detection}}{\# \text{ of attempt}} \times 100, \tag{12}$$

$$\text{false positive ratio}(\%) = \frac{\# \text{ of incorrect object detection}}{\# \text{ of attempt} \times \# \text{ of objects in DB}} \times 100, \tag{13}$$

$$\text{false negative ratio}(\%) = \frac{\# \text{ of correct object detection fail}}{\# \text{ of attempt}} \times 100. \tag{14}$$

Here, the more objects the database contains, the more false positive recognition occurs. Therefore, we normalize the false positive ratio by the number of objects of in the database.

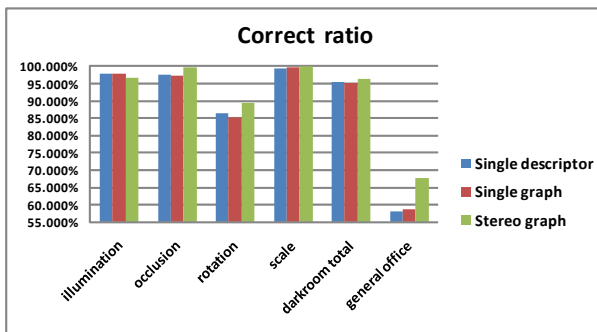


Fig. 5. Overall correct ratio comparison under the condition variations

Although the overall correct ratio of the proposed methods under the dark-room environments is similar with that of using descriptor matching, the false positive ratio of graph-based methods decreases compared to the descriptor

Table 1. Performance of three methods for 2880 images of 80 objects in two environments - correct, false positive, and false negative ratio

variation	Single descriptor			Single graph			Stereo graph		
	correct	false +	false -	correct	false +	false -	correct	false +	false -
darkroom	95.347	0.377	4.653	95.000	0.238	5.00	96.389	0.257	3.611
general	58.056	0.304	41.944	58.611	0.204	41.389	67.778	0.165	32.222

Table 2. Performance for 2880 images of 80 objects in controlled darkroom environment and general office environment - correct and false positive ratio. Without verification, we selected the best object from the graph matching results.

variation	Single descriptor matching		Single graph matching		Single stereo matching	
	correct	false +	correct	false +	correct	false +
darkroom	98.160	1.840	98.750	1.250	98.264	1.736
general	63.056	36.944	66.944	33.056	70.556	29.444

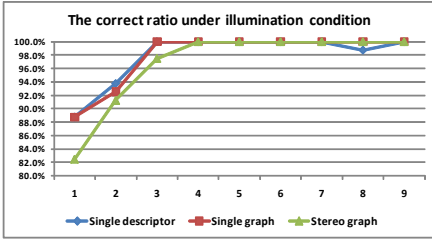
matching method. In general office environments, the performance of graph-based methods is much more improved. In a single image case, although the correct ratio of a graph-based method is similar with that of a descriptor matching method, the false positive ratio of a graph-based method decreases compared to descriptor matching method. In a stereo image case, both the correct ratio and the false positive ratio are improved. The correct ratio of a graph-based method increases relatively 16.7% compared to the descriptor matching method, and the false positive ratio of a graph-based method decreases relatively 45.7 % compared to the descriptor matching method.

The performance under the special condition variations is shown in Fig. 6.

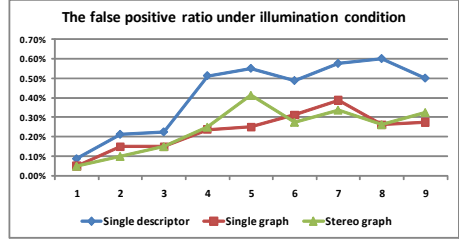
We can see that the false positive ratio of graph-based methods is generally less than that of the descriptor matching method. Especially, the correct ratio of the stereo graph-based method is better than the others under the occlusion and yaw condition variations.

In fact, absolute evaluation may be meaningless because the performance can vary with the database used, the types of selected local features, and the definition of correct ratio and false positive ratio. However, it is worthy of note that the performance when using graph-based representation and matching methods (both single and stereo cases) is better than when using a simple descriptor matching method. Especially, the performance for general situations is greatly improved. This shows that the recognition using graph-based object representation and matching is more robust under photometric and geometric changes while greatly reducing inherent ambiguity.

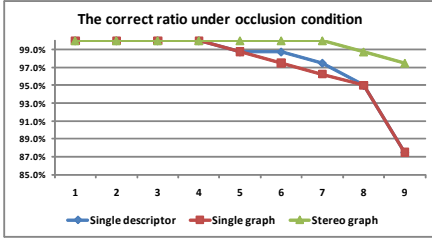
The result in Table 2 is obtained just by selecting one best candidate based on the descriptor or graph matching without verification. In this case, we do not compute the false negative ratio. These results also show that the graph-based methods are better than the descriptor matching method.



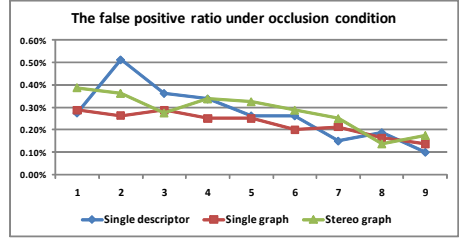
(a) Correct under illumination variation



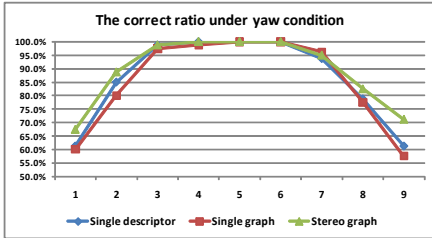
(b) False positive under illumination variation



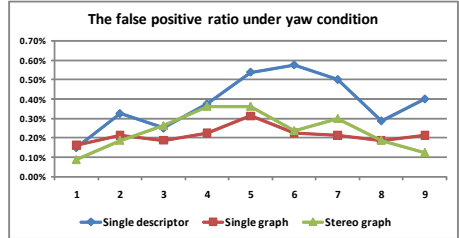
(c) Correct under occlusion variation



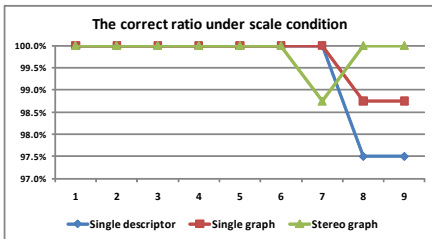
(d) False positive under occlusion variation



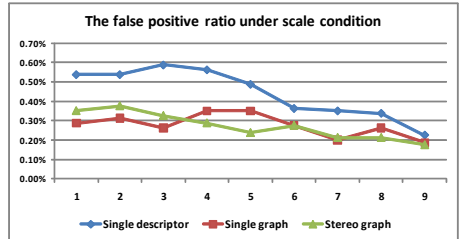
(e) Correct under yaw variation



(f) False positive under yaw variation



(g) Correct under scale variation



(h) False positive under scale variation

Fig. 6. Performance under the condition variations. Refer to the percentage on the left.

Through the experiments, it is clearly shown that the recognition using relational information is more robust and less ambiguous than the recognition based on the simple local feature matching under photometric and geometric variations. However, at the same time, it needs more computation time. While the

computation time is proportional to the number of local features in the individual feature-matching based approach, it is proportional to both the number of local features and the number of graphs (i.e., number of objects in the database) in the proposed method. In our experiments, when the database contains 100 objects, the graph-based method takes 77.21 seconds for a single image and 78.24 seconds and for a stereo image pair while the simple feature-matching based method takes 2.62 seconds on average.

7 Conclusion

In this paper, we have presented a new method for texture object recognition, aiming at reducing inherent ambiguity due to ambiguous local features. To this end, unlike simple descriptor matching methods, we use an attributed relational graph (ARG) represent an object with local features and their relational information together.

We generate proper object ARGs and an input ARG in accordance with a single image and a stereo image. Next, we select appropriate candidate objects via inexact graph matching. Finally, we perform verification with homography, fundamental matrix, or 3D rigid transformation. The proposed framework shows promising performance for a single image and a stereo image. In addition, it shows robust recognition performance under cluttered scene. However, it needs more computation time. Therefore, we will try to develop faster algorithms as well.

Acknowledgement. This work has been partially supported by Samsung Electronics in 2009.

References

1. Lindeberg, T.: Detecting salient blob-like image structures and their scales with a scale-space primal sketch: a method for focus-of-attention. *International Journal of Computer Vision* 11, 283–318 (1993)
2. Shokoufandeh, A., Marsic, I., Dickinson, S.: View-based object matching. In: *IEEE International Conference on Computer Vision*, pp. 588–595 (1998)
3. Schmid, C., Bauckhage, R.M.: Evaluation of interest point detectors. *International Journal of Computer Vision* 37, 151–172 (2000)
4. Mikolajczyk, K., Schmid, C.: Scale & affine invariant interest point detectors. *International Journal of Computer Vision* 60, 63–86 (2004)
5. Ke, Y., Sukthankar, R.: Pca-sift: a more distinctive representation for local image descriptors. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 506–513 (2004)
6. Kim, S., Yoon, K.J., Kweon, I.S.: Object recognition using a generalized robust invariant feature and gestalt's law of proximity and similarity. *Pattern Recognition* 41, 726–741 (2008)
7. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60, 91–110 (2004)

8. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 1615–1630 (2005)
9. Tico, M., Kuosmanen, P.: Fingerprint matching using an orientation-based minutia descriptor. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25, 1009–1014 (2003)
10. Schmid, C.: A structured probabilistic model for recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 485–490 (1999)
11. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 264–271 (2003)
12. Moreels, P., Maire, M., Perona, P.: Recognition by probabilistic hypothesis construction. In: *Eurographics Symposium on Rendering*, pp. 55–68 (2004)
13. Ballard, D.H.: Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition* 13, 111–122 (1981)
14. Lamdan, Y., Wolfson, H.J.: Geometric hashing: A general and efficient model-based recognition scheme. In: *International Conference on Computer Vision*, pp. 238–249 (1988)
15. Dorkó, G., Schmid, C.: Selection of scale-invariant parts for object class recognition. In: *International Conference on Computer Vision*, pp. 634–639 (2003)
16. Sivic, J., Russell, B.C., Efros, A.A., Zisserman, A., Freeman, W.T.: Discovering object categories in image collections. In: *International Conference on Computer Vision* (2005)
17. Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structure for object recognition. *International Journal of Computer Vision* 61, 55–79 (2005)
18. Crandall, D., Felzenszwalb, P., Huttenlocher, D.: Spatial priors for part-based recognition using statistical models. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10–17 (2005)
19. Gold, S., Rangarajan, A.: A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18, 377–388 (1996)
20. Caetano, T.S., Cheng, L., Le, Q.V., Smola, A.J.: Learning graph matching. In: *IEEE International Conference on Computer Vision*, pp. 1–8 (2007)
21. Torresani, L., Kolmogorov, V., Rother, C.: Feature correspondence via graph matching: Models and global optimization. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part II. LNCS*, vol. 5303, pp. 596–609. Springer, Heidelberg (2008)
22. Graciano, A., Cesar Jr., R., Bloch, I.: Graph-based object tracking using structural pattern recognition. In: *Proc. of SIBGRAPI*, pp. 179–186 (2007)
23. Carneiro, G., Jepson, A.D.: Object recognition using flexible groups of local features. Technical Report CSRG-481 (2004)
24. Conte, D., Foggia, P., Sansone, C., Vento, M.: *How and why pattern recognition and computer vision applications use graph*. Springer, New York (2007)
25. Guerra Filho, G.: Disambiguating the recognition of 3d objects. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2278–2285 (2009)
26. Leung, T., Malik, J.: Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision* 43, 29–44 (2001)

Posing to the Camera: Automatic Viewpoint Selection for Human Actions

Dmitry Rudoy and Lihi Zelnik-Manor

Technion, Haifa, Israel

Abstract. In many scenarios a scene is filmed by multiple video cameras located at different viewing positions. The difficulty in watching multiple views simultaneously raises an immediate question - which cameras capture better views of the dynamic scene? When one can only display a single view (e.g. in TV broadcasts) a human producer manually selects the best view. In this paper we propose a method for evaluating the quality of a view, captured by a single camera. This can be used to automate viewpoint selection. We regard human actions as three-dimensional shapes induced by their silhouettes in the space-time volume. The quality of a view is evaluated by incorporating three measures that capture the visibility of the action provided by these space-time shapes. We evaluate the proposed approach both qualitatively and quantitatively.

1 Introduction

With the advances of recent years video cameras can now be found in abundance. Scenes and events are frequently being recorded not only by a single camera, but rather by multiple ones, e.g., school children sports events are recorded by many eager parents. To visualize such data on a single screen one needs to select the single "best" camera for each moment in the event. In movie and TV production the "best" camera view is selected manually by the producer. In non-professional scenarios, however, such a producer is typically not available. We are therefore interested in automating the process of camera selection.

Automatic viewpoint selection has been addressed before for 3D computer gaming and graphics applications [6]. Several methods, e.g., [17,19,5] have been proposed for optimal view selection for static 3D objects. Assa et al. in [1,2] proposed methods for camera control when viewing human actions. These solutions, however, rely on knowing the 3D structure of the scene and, hence, are not applicable to real-world setups filmed by video cameras. Camera selection has also been previously explored for surveillance applications, however, there the camera setup and the goal are typically different from ours. Gorshorn et al. [10] propose a camera selection method for detecting and tracking people using camera clusters. Most of the cameras in the cluster do not overlap and hence the main goal is tracking the person as he moves from one camera view to the other. In [15,7] methods were proposed for selecting the camera that provides the best view for recognizing the identity of a viewed person. This requires mostly face visibility.

In this paper we propose a technique for video-based viewpoint quality evaluation for actions. We show that ranking camera views according to the action visibility they provide is useful for selecting the best view to display. We first discuss what makes one view better than the other. Our guiding principle is that the better views are those where the action is easier to recognize, since the limbs and their motion are clearly visible. To detect views with good limb visibility we propose three measures (spatial, temporal, and spatio-temporal), which capture the properties of the preferable views. Finally, we incorporate the spatial, temporal and spatio-temporal measures into a single global score.

The usefulness of the proposed approach is evaluated qualitatively on real video data of sports events, dance and basic human actions. Additionally we test our approach on 3D gaming scenarios. To provide some quantitative evaluation we further test the usefulness of the proposed approach for action recognition. Our experiments show that selecting a single view to process does not degrade recognition, but rather the opposite occurs and recognition rates are improved. This could speed-up recognition in multi-camera setups.

The contribution of the paper is hence threefold. First, it presents several video-based properties of preferable views of human actions (Section 2). Second, a method is proposed for capturing these properties and hence estimating the relative quality of the views (Section 3). Last, we demonstrate the benefits of the suggested approach in scene visualization and establish the selection of the better views by action recognition (Section 4).

2 Why Some Views Are Better than Others?

Many human actions are easier to recognize from some viewpoints, compared to others, as illustrated in Fig. 1. This is why the "WALK" road-sign always shows the stride from the side. Searching YouTube for "golf swing" retrieves almost only front views and no back or diagonal views. Searching for "hugging people" yields almost only side views showing both people approaching each other. The mutual to these examples is that generally the better views are those showing the limbs and their motion clearly. For the time being we limit our analysis to scenes showing a single person performing free actions. Later on, in Section 4.1 we discuss how multi-player scenes are handled.

Our goal is to evaluate limb visibility without tracking the limbs, since limb detection is time consuming and error prone. We achieve this by observing that good visibility of the limbs and their motion has generic temporal, spatial and spatio-temporal implications on the space-time shape induced by the silhouettes (as illustrated in Fig. 2):

1. Temporal: High motion of the limbs implies that the silhouettes vary significantly over time.
2. Spatial: Good visibility of the limbs implies that the outlines of the silhouettes are highly concave. For example, the spread out legs in a side view of walk generate a large concavity between them.



Fig. 1. Many actions are better captured from a specific view point. Walking and hugging are best captured from the side, while a golf swing is best viewed from the front. Top row: examples of road signs. Bottom rows: YouTube search results for "hugging people" and "golf swing".

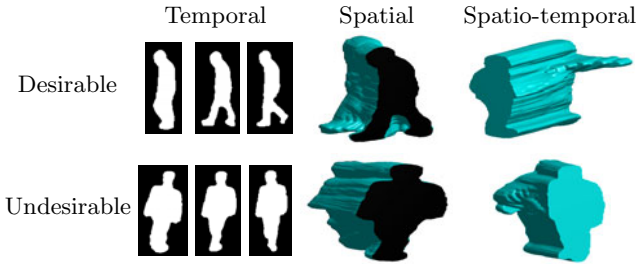


Fig. 2. Properties of good views. Good views are those where the limbs and their motion are clearly visible. Temporally this implies large variations (left). Spatially, the silhouettes in a good view are highly concave (middle). In space-time visibility of the limbs and their motion implies shapes with significant saliency (right).

3. Spatio-temporal: When the limbs and their motion are clearly visible the resulting space-time shape is not smooth, but rather has protruding salient parts (corresponding to the moving limbs). Conversely, self occlusions and lack of motion lead to smooth space-time shapes.

Interestingly, each of these three properties matches known properties of human perception. First, it is known that human vision is attracted to high motion [12]. This corresponds to the temporal property. Second, in 1954 Attneave [3] proposed that the most informative regions along a visual contour are those with

high curvature. Extending this idea to three dimensions matches the spatio-temporal property, that looks for protruding regions with high curvature in space-time. Finally, Feldman and Singh showed in [8] that for closed planar contours concave segments carry greater information, than corresponding convex ones. Correspondingly, the presented spatial property captures the concavity of the silhouettes.

3 Evaluating Viewpoint Quality

In this section we propose a method for viewpoint quality evaluation, which is based on the principles presented above. We intentionally seek simple measures of viewpoint quality to enable fast processing.

3.1 Measures of Action Visibility

Following [9] we regard human actions as three-dimensional shapes induced by their silhouettes in the space-time volume. Since cameras at different positions view different aspects of the scene, the silhouette extraction can vary between cameras. Our measures do not require perfect silhouettes, however, we do assume the silhouettes are acceptable, i.e., when there are no self occlusions the limbs should be visible in the silhouette. This assumption is reasonable in many scenarios, e.g., computer games, day-time sports events and security setups where the background can be modeled accurately and does not change. We compensate for the global translation of the body in order to emphasize motion of parts relative to the torso. This is done by aligning the centers of mass of the silhouettes. Note, that cameras at different positions view different silhouettes, hence, the induced space-time shapes are different. We assume that all the cameras view the person fully without external occlusions. Self occlusions (e.g., as in a top view) are allowed.

Spatio-temporal measure: Shape saliency. In accordance with property (3), when the limbs are visible the induced space-time shape is not smooth. To build a spatio-temporal measure of action visibility we need to quantify the unsmoothness of the space-time shapes. We base our approach on the method proposed by Lee et al. [16] for evaluation of saliency and viewpoint selection for 3D meshes. Their work proposes a method for measuring saliency at every vertex of a static 3D mesh. Saliency is defined as the deviation of the mesh from a perfectly smooth shape, i.e., sphere. Furthermore, they propose to evaluate a viewpoint quality by summing up all the saliency values of all the visible parts from a given viewpoint. In our work instead of the same shape viewed from different directions we have a different shape for each view. Thus measuring the overall saliency of the space-time shape allows us to estimate the quality of a view, that produced that shape.

Following [16] we first calculate the local space-time saliency at each point on the shape's surface. This is captured by the difference between the point's local curvature at different scales. Then we evaluate global saliency by summing all the local saliency values, since every point on the surface of the space-time shape

is visible. The method in [16] was limited to 3D meshes. In our case, however, the shapes are represented in voxels and not meshes. We next follow the ideas of [16] and extend them to voxel-base representations of space-time shapes.

To compute the local saliency of points on the surface of the space-time shape we first calculate the mean curvature $\kappa_m(p)$ of each surface point p using the method proposed by Kindlmann et al. [14]. Next, following [16] we define the weighted mean curvature, $G(\kappa_m(p), \sigma)$, at each space-time shape surface point, as:

$$G(\kappa_m(p), \sigma) = \frac{\sum_q \kappa_m(q) W(p, q, \sigma)}{\sum_q W(p, q, \sigma)} \quad (1)$$

where the sum is over all the points q in a 2σ radius neighborhood around point p and $W(p, q, \sigma)$ is a weight function. Note, that as opposed to the 3D models used in computer graphics, our shapes can have different scales in space and in time. Hence, we define $W(p, q, \sigma)$ as:

$$W(p, q, \sigma) = \exp\left(-\frac{1}{2} \left(\sum_{j \in \{x, y, t\}} \frac{(p_j - q_j)^2}{\sigma_j^2} \right)\right) \quad (2)$$

where $p = (p_x, p_y, p_t)$ and $q = (q_x, q_y, q_t)$ are two points on the space-time shape surface, and $\sigma = (\sigma_x, \sigma_y, \sigma_t)$. Local space-time saliency is then defined as the absolute difference between two weighted curvatures:

$$L(p) = |G(\kappa_m(p), \sigma) - G(\kappa_m(p), 2\sigma)| \quad (3)$$

Lee et al. [16] propose to incorporate multiple scales, but in our space-time shapes this does not bring the desired effect. This is because 3D models usually have many small details, that need to be taken into account. In contrast, our space-time shapes have little detail, thus, it would suffice to find a single optimal value of σ corresponding to a single scale. We have experimentally selected $\sigma = (5, 5, 3)$. This value emphasizes the unsmoothness of the shapes in the best way while giving low saliency values to the flat regions. It was kept fixed for all the results presented here.

Figure 3 shows the local space-time saliency values of Eq. (3) for all the surface points of the space-time shape of a real punch action obtained from different views. It can be seen that the moving parts, e.g., the arm, generate high curvature surfaces and hence receive high local saliency values, while stationary parts, e.g., the legs, which imply flat areas in the space-time shape, receive low local values.

Finally, we define the spatio-temporal visibility score of a view as the sum of the local saliency values of all the points on the surface of the space-time shape:

$$S_{ST} = \sum_p L(p) \quad (4)$$

The values of the spatio-temporal saliency score S_{ST} for a punch action are also marked in Fig. 3. We note here that S_{ST} is not bound from above, however it is always non-negative.

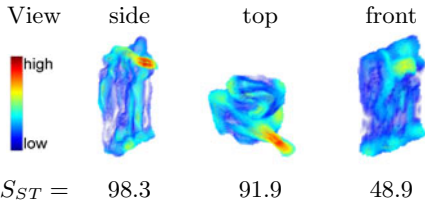


Fig. 3. Spatio-temporal visibility measure. Local saliency values for space-time shapes obtained from different views of the same punch action nicely emphasize the more important regions, in this case, the punching arm. The side and top views show the protruding punching arm and hence their total visibility score S_{ST} is high. The front view produces a smooth shape and correspondingly a low S_{ST} .

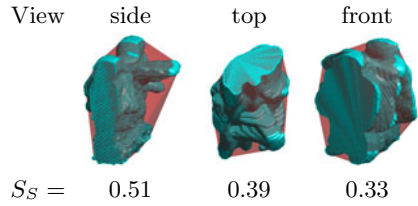


Fig. 4. Spatial visibility measure. An illustration of a space-time shape (cyan) and its convex hull (red) for a "punch" action captured from different angles. The side view receives the highest S_S score since it shows more concave regions under the arm. The top and front views, where the limbs are less visible, obtain lower scores, since their outlines are more convex.

Spatial measure: Visibility of limbs. According to property (2), when the limbs are fully visible the outlines of the induced silhouettes are highly concave. To quantify how concave a shape is we seek for a simple and fast measure. One such measure is computing the volume difference between the 3D convex hull of the space-time shape and the shape itself. We define the spatial measure as:

$$S_S = 1 - \frac{V_{sh}}{V_{ch}} \quad (5)$$

where V_{sh} is the volume of the space-time shape and V_{ch} is the volume of its convex hull. The S_S score is non-negative and bounded by 1 from above. Figure 4 illustrates some space-time shapes together with the corresponding 3D convex hulls.

Temporal measure: Detecting large variations. Following property (1), we wish to discover views which exhibit a significant pattern of change along time. We measure this by computing the portion of pixels where motion was observed somewhere along the sequence. Note, that we do not care what was the type of motion or when it occurred. Our only interest is the amount of motion. Since we would like our measures to be as simple and fast to compute as possible, we evaluate the amount of motion as follows. Let $g(x, y, t)$ be the silhouette indicator function and τ be the temporal length of the sequence. Following 4 we build the motion-energy image describing the sequence as:

$$E(x, y) = \bigcup_{t=0}^{\tau-1} g(x, y, t) \quad (6)$$

We denote by $g_m(x, y)$ the biggest single-frame silhouette (in sense of number of pixels) in the sequence. The temporal measure is then defined as:

$$S_T = 1 - \frac{\sum_{x,y} g_m(x, y)}{\sum_{x,y} E(x, y)} \quad (7)$$

Note, that this score is always non-negative and bounded by 1 from above.

To illustrate the temporal motion-based score we present in Fig. 5 $E(x, y)$, $g_m(x, y)$ and S_T for different views of a kick action. As can be seen, the side view, where the action is better viewed, presents a higher percentage of moving pixels (gray), and thus receives a higher score.

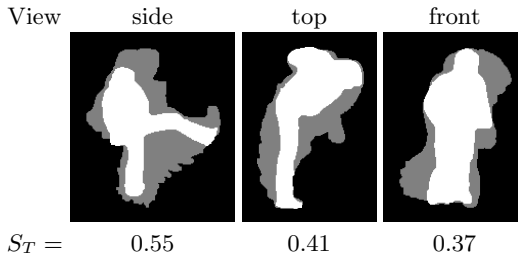


Fig. 5. Temporal visibility measure. The motion-energy image E (gray) superimposed on the biggest silhouette g_m (white) for a "kick" action seen from different views. The side view captures the leg motion in full and hence receives the highest score, while the front view shows very little of the leg motion and hence receives a low score.

3.2 Combining Visibility Measures

In the previous section we presented three different visibility measures, which capture somewhat different notions. To combine them into a single unified measure without parameter tuning we take the product of the three. Our final view quality measure is hence:

$$S = S_{ST} \cdot S_S \cdot S_T \quad (8)$$

4 Applications and Experiments

In this section we demonstrate the usefulness of the proposed approach for simplified visualization of multi-camera scenes (Section 4.1). We cannot compare our view selection method to previous works, like [1,2], since they use 3D scene data, which is not available in our case, hence the evaluation is mostly qualitative. To provide some quantitative evaluation, we further show that selection of a single camera does not harm action recognition and in some cases even improves the performance rate (Section 4.2).

4.1 Automatic Camera Selection

As discussed earlier, a relevant application of the proposed viewpoint quality estimation method is automatic selection of the best camera. Given multiple

video streams of the same scene pictured from different points of view, our goal is to produce a single video showing each action from its preferred viewpoint.

The quality of the view provided by a certain camera can change with time, depending on the performed actions and the person's orientation in space. To take those changes into account we experimented with two methods. The first splits the input videos into short non-overlapping sub-sequences (of 36 frames), computes the view quality S for each of them and then selects the best view for each set of corresponding sub-sequences. This method is fast, however, changes in the selected viewpoint do not always occur at the optimal moments due to the a-priori selection of temporal cuts.

The second approach is somewhat slower, however its results are more accurate. Rather than splitting the videos we adopt a sliding-window approach where view selection is applied to all sets of corresponding sub-sequences of length 36 frames. This yields 36 independent “best-view” decisions for each set of corresponding frames. For each frame the view which received the majority of votes is selected. To avoid redundant view switches we accept a view change only when it lasts longer than 25 frames, i.e., we do not switch to a new camera if it does not provide good visibility for more than approximately one second.

We begin by testing the proposed framework on a golf scene. We have intentionally selected golf since googling for “golf swing” videos retrieves many tutorials, most of them show the swing from the same frontal viewpoint, thus making it somewhat clear what is the desired result. In our setup eight cameras viewed a golfer hitting the ball four times, each time rotating to face a different

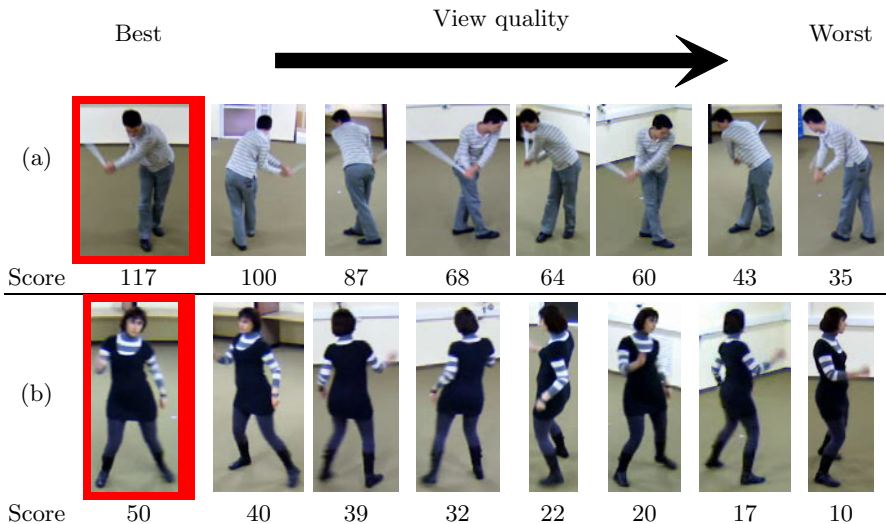


Fig. 6. View selection for golf swing and dance. Views showing clearly all the limbs together with their motion are ranked higher while views with severe self-occlusion are detected as low quality.

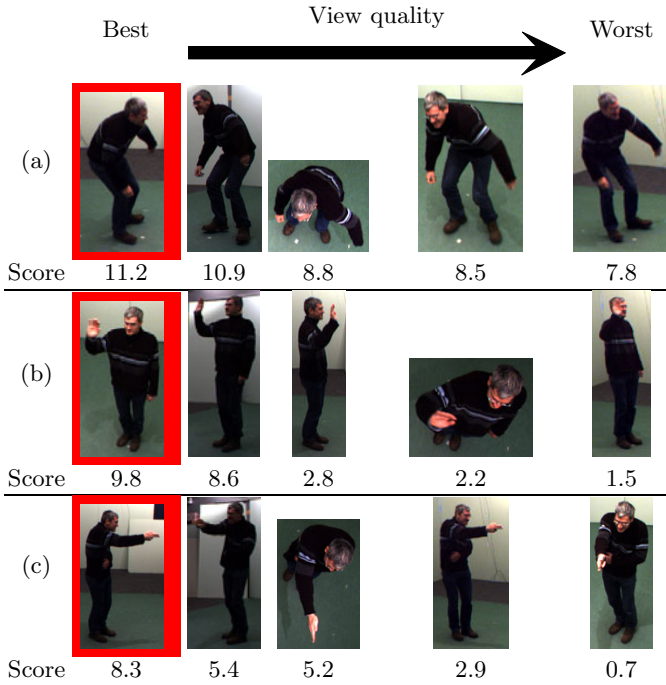


Fig. 7. View selection on IXMAS. Examples of view selection applied to IXMAS sequences of a single person. Top row (a) shows a "get-up" action, which is visible clearly from any angle, thus the view qualities do not vary too much. Rows (b) and (c) show actions where some views are preferred. In this case the different views are nicely rated according to the visibility they provide. Views showing clearly all limbs are ranked best while views with severe self-occlusion are ranked worst.

camera. As shown in Fig. 6 (a) and in the supplemental video¹ our view selection approach successfully selects the frontal view for the swing, in line with what is used for golf tutorials.

Next we test the framework on a simple dance move. This move is best viewed from front, but the actress repeats it several times, each time facing a different direction. As in the golf scene, our view selection approach clearly prefers the front view. Other views are ranked according to the visibility of the limbs motion, as shown in Fig. 6 (b).

We further applied the proposed view selection to the IXMAS database [11], which includes 12 actors performing 13 everyday actions continuously. Each actor performs the set of actions three times, and the whole scene is captured by 5 synchronized video cameras (4 side cameras, that cover almost half a circle around the subject and one top camera). The actors selected freely their orientation, hence, although the cameras were fixed, each viewed the actors from varying angles. In other words, we cannot label a certain camera as front view

¹ The supplemental video is available at <http://webee.technion.ac.il/labs/cgm/>

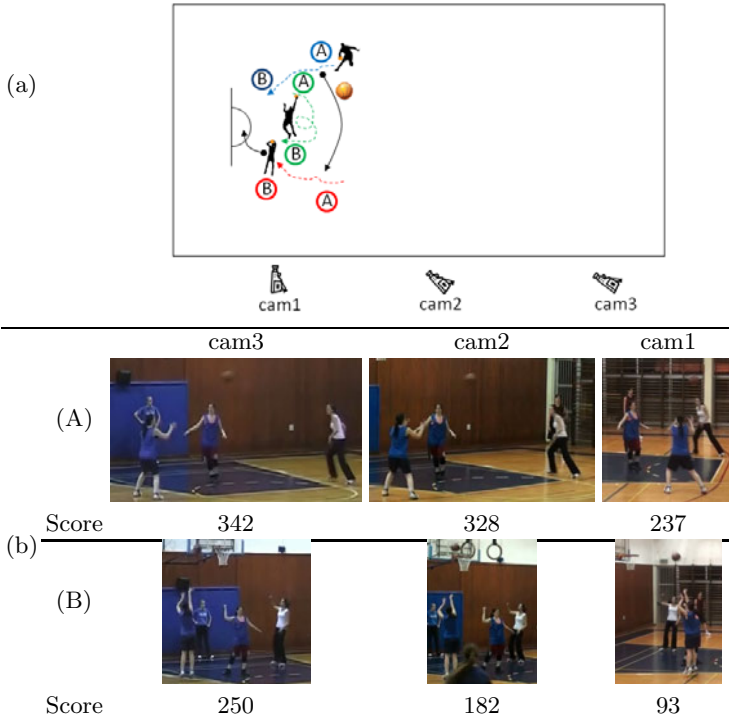


Fig. 8. (a) A sketch of the filming setup of a basketball drill. The blue, green and red curves illustrate the paths of the players. The labels A and B mark the location in the field of each player at moments A and B. (b) Example frames from the three cameras at moments A and B depicted in the sketch. Camera 1 received the lowest rates since the arm motion in throwing the ball is occluded in (A) and the players occlude each other in (B). Camera 2 received higher rates since there are less self occlusions. Camera 3 got the highest view quality rates since there are no occlusions and the arm motion is clearly visible both in (A) and in (B). Please view the supplemental video.

since it captured both front and side views. In this experiment we do not use the ground-truth provided with the database. As can be seen in Fig. 7 and in the supplemental video, our system consistently selects views where the action performed by the person is clearly visible. For example, for walking the algorithm selects the side view with the maximum visibility of the moving legs, and for waving the front view is selected, such that the hand motion is clearly visible.

To show the applicability of the proposed view selection method to more challenging videos we filmed our local basketball team during training using three fixed cameras. The cameras were set along the court, as shown on Fig. 8 (a). In this scene, three players performed a drill that included running and free throws. We extracted the players' silhouettes for each camera using simple background subtraction. This led to very noisy silhouettes. Furthermore, in significant parts of the scene the players were either very close to each other or occluded each other. Thus it was not practical to treat each player independently. Instead, we applied



Fig. 9. A single player throwing a ball viewed from 3 viewpoints. As expected, the side view gets the highest rate.

our view quality estimation to the joint shape of all three players, as if they were a single subject. As can be seen in Fig. 8(b), camera 1 suffers from severe occlusions, camera 2 suffers from partial occlusions, while camera 3 captures most of the drill without occlusions. Despite the fact that the spacial measure is less informative in this case, our view quality rating reflects that nicely.

Figure 9 shows results of a single-player scene, throwing a ball. Here our approach nicely detects the side view, where the throwing of the ball is best captured. These results demonstrate that the proposed view quality rating can be applied to single and multi-player scenes as one.

Additionally to the videos taken by real video cameras, our viewpoint quality estimation is also relevant in 3D graphics, or specifically in 3D games. To show the applicability of the proposed method in this field we choose a scene of two hugging people from Sims 3 game, filmed from eight different angles. Note, that in this case perfect silhouettes are available. Since the figures touch each other most of the action, we treated them as a single subject and applied the viewpoint quality of Eq. (8). As shown in Fig. 10, the side views get a higher ranking while the front and back views, with severe self occlusions, are least preferred. This matches what one typically expects to see in "hugging people" videos.

4.2 Action Recognition

As far as we know, there are no databases with ground-truth view selection, hence quantitative evaluation is somewhat difficult. To demonstrate that our technique selects good views we test its performance as a pre-processing step for action recognition. Given an action filmed from multiple angles we select a single view using the quality measure of Eq. (8). We then classify this view only. Our experiments show that selecting a single view does not harm the recognition rates, implying that the views we select are good for recognition.

We test this framework on the IXMAS database [11]. We split the long videos according to the provided ground-truth, into shorter action clips, so that each shows a single action. Following previous work we test on a leave-one-actor-out scenario, i.e., we take all the performances of one actor as the testing set and all other actors as the training set. In our experiments we used only 10 actors (excluding Srikumar and Pao) and 11 actions (excluding "throw" and "point") for fair comparison with previous work, which excluded these as well. For each

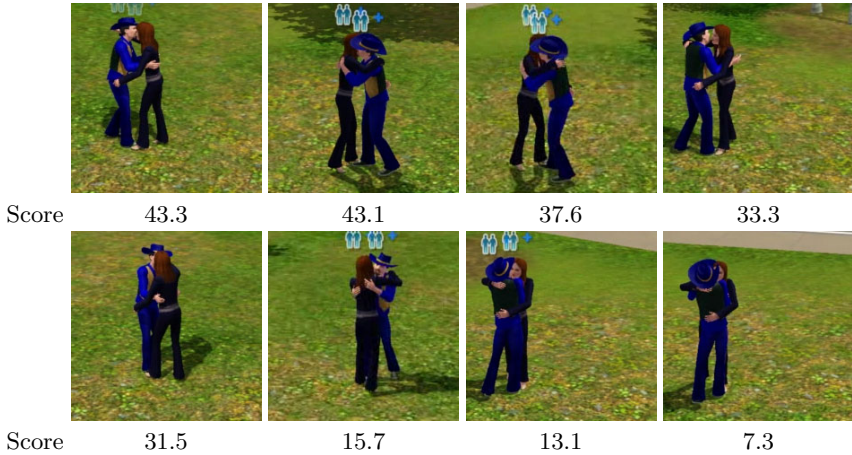


Fig. 10. View quality estimation for hug action in Sims 3 game. The different views are nicely rated according to the visibility they provide. Views where the interaction is clearly visible are ranked better than views in which one of the figures is occluded.

clip in the test set we evaluate the viewpoint quality provided by each camera using the proposed measure of Eq. (8). Then we classify the action in the view with the highest score using one of three monocular recognition methods: (i) the silhouette based approach of Gorelick et al. [9]² (ii) the view invariant approach of Junejo et al. [13]³ and (iii) the silhouette and optical flow based approach of Tran and Sorokin [18]⁴

For every method we compare the results of the recognition after view selection with three other options: (i) average recognition rate, which reflects random selection of views, (ii) the rate of the "best" camera and (iii) the rate of the "worst" single camera. In "best" / "worst" camera we refer to the single camera with the highest / lowest recognition rate. In practice, selecting the "best" camera is not feasible, since it requires an oracle that a-priori tells us which of the fixed views will be better. However, this is the best rate that can be achieved from a single fixed camera. On contrary, a wrong selection of the camera could lead to "worst" camera rates. Table 1 shows that the proposed view selection either comparable, or improves the results of the "best" camera. This implies that the selected views are those where the action is recognizable, which satisfies the goal of this work. It is further interesting to note that the "best" fixed camera is different for each recognition method. This implies that on average different methods have preference for different viewpoints. Nevertheless, our view selection succeeds in detecting those views which are recognizable by all methods.

² For [9] we obtained code from the authors.

³ For [13] we used our own implementation which obtains results similar to those reported in the original paper.

⁴ For [18] we used authors' code available at their website with 1NN classifier. However, we used a slightly different experimental setup, thus yielding slightly different results.

Table 1. Comparison of recognition rates for different recognition methods shows that the proposed view selection performed before recognition often improves the best fixed camera rate. Note that an a-priori selection of the "best" camera is not possible. We mark in parentheses the label of the fixed camera that turned out to provide the best/worst recognition rates. Note that for each recognition method the best performance was obtained with a different camera.

	Proposed view selection	Average	Best fixed camera (selected a-posteriori)	Worst fixed camera (selected a-posteriori)
9	81	73	77 (cam4)	63 (cam5)
18	89	85	88 (cam3)	82 (cam5)
13	65	64	67 (cam1)	57 (cam5)

5 Conclusion

This paper presented a method for viewpoint quality estimation of human actions. To determine better views we compute a visibility score based on properties of the space-time shape induced by the actor's silhouettes. Our experiments show that the proposed approach can successfully estimate the action visibility provided by each camera. Such estimation can be used for automatic selection of a single best view of one or more actors. Furthermore, selecting the best view of the action prior to the recognition improves the rates of the monocular action recognition method, together with speeding them up (since we need to recognize only one view).

Acknowledgments

This research is supported by Marie Curie IRG-208529.

References

1. Assa, J., Cohen-Or, D., Yeh, I., Lee, T.: Motion overview of human actions. In: International Conference on Computer Graphics and Interactive Techniques. ACM, New York (2008)
2. Assa, J., Wolf, L., Cohen-Or, D.: The Virtual Director: a Correlation-Based Online Viewing of Human Motion. In: Eurographics (2010)
3. Attneave, F.: Some informational aspects of visual perception. *Psychological review* 61, 183–193 (1954)
4. Bobick, A.F., Davis, J.W.: The recognition of human movement using temporal templates. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 23, 257–267 (2001)
5. Bordoloi, U., Shen, H.: View selection for volume rendering. *IEEE Visualization* 5, 487–494 (2005)
6. Christie, M., Olivier, P., Normand, J.: Camera control in computer graphics. *Computer Graphics Forum* 27, 2197–2218 (2008)

7. El-Alfy, H., Jacobs, D., Davis, L.: Assigning cameras to subjects in video surveillance systems. In: Proceedings of the 2009 IEEE International Conference on Robotics and Automation, pp. 3623–3629. Institute of Electrical and Electronics Engineers Inc. (2009)
8. Feldman, J., Singh, M.: Information along contours and object boundaries. *Psychological Review* 112, 243–252 (2005)
9. Gorelick, L., Blank, M., Shechtman, E., Irani, M., Basri, R.: Actions as Space-Time Shapes. *Transactions on Pattern Analysis and Machine Intelligence* 29, 2247–2253 (2007)
10. Goshorn, R., Goshorn, J., Goshorn, D., Aghajan, H.: Architecture for cluster-based automated surveillance network for detecting and tracking multiple persons. In: 1st Int. Conf. on Distributed Smart Cameras, ICDS-C (2007)
11. (IXMAS), <http://charibdis.inrialpes.fr>
12. Johansson, G.: Visual perception of biological motion and a model for its analysis. *Perceiving Events and Objects* (1973)
13. Junejo, I., Dexter, E., Laptev, I., Pérez, P.: View-Independent Action Recognition from Temporal Self-Similarities. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2010)
14. Kindlmann, G., Whitaker, R., Tasdizen, T., Moller, T.: Curvature-based transfer functions for direct volume rendering: Methods and applications. In: Proceedings of the 14th IEEE Visualization 2003, vol. 67. IEEE Computer Society, Los Alamitos (2003)
15. Krahnstoeber, N., Yu, T., Lim, S., Patwardhan, K., Tu, P.: Collaborative Real-Time Control of Active Cameras in Large Scale Surveillance Systems. In: Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications-M2SFA2 (2008)
16. Lee, C., Varshney, A., Jacobs, D.: Mesh saliency. *ACM Transactions on Graphics* 24, 659–666 (2005)
17. Mudge, M., Ryan, N., Scopigno, R.: Viewpoint quality and scene understanding. In: *Vast 2005*, p. 67 (2005)
18. Tran, D., Sorokin, A.: Human activity recognition with metric learning. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part I*. LNCS, vol. 5302, pp. 548–561. Springer, Heidelberg (2008)
19. Vazquez, P., Feixas, M., Sbert, M., Heidrich, W.: Automatic view selection using viewpoint entropy and its application to image-based modelling. *Computer Graphics Forum* 22, 689–700 (2003)

Orthogonality Based Stopping Condition for Iterative Image Deconvolution Methods

Dániel Szolgay¹ and Tamás Szirányi²

¹ Pázmány Péter Catholic University, Budapest, Hungary
szoda@digitus.itk.ppke.hu

² Computer and Automation Research Institute, MTA SZTAKI, Budapest, Hungary
sziranyi@sztaki.hu

Abstract. Deconvolution techniques are widely used for image enhancement from microscopy to astronomy. The most effective methods are based on some iteration techniques, including Bayesian blind methods or Greedy algorithms. The stopping condition is a main issue for all the non-regularized methods, since practically the original image is not known, and the estimation of quality is based on some distance between the measured image and its estimated counter-part. This distance is usually the mean square error (MSE), driving to an optimization on the Least-Squares measure. Based on the independence of signal and noise, we have established a new type of error measure, checking the orthogonality criterion of the measurement driven gradient and the estimation at a given iteration. We give an automatic procedure for estimating the stopping condition. We show here its superiority against conventional ad-hoc non-regularized methods at a wide range of noise models.

1 Introduction

In almost all image acquisition processes blurring is a common issue. Due to various reasons (like defocusing, atmospheric perturbations, optical aberrations, motion), the acquired images are distorted, and sometimes without restoration, useless. The distortion is often modeled as convolution: the original unknown image is convolved with a so called Point Spread Function (PSF) that describes the distortion, that a theoretical point source of light takes through the image acquisition process.

$$Y = H * U + N \quad (1)$$

where Y is the measured blurry image, U is the unknown original image, H is the PSF and N is additional noise. Y , U and N are (n, m) sized 2D images and H is a (k, l) sized kernel ($k \leq n, l \leq m$).

In the last decades a lot of methods were developed in order to restore the original image from the blurred, noisy measurement. The methods aiming to eliminate the effect of the convolution are called deconvolution methods. They can be classified based on different aspects, like linear/non-linear, iterative/non-iterative, the assumed noise model, etc. (see [1] for details).

In some cases we can assume that the PSF is known a priori [2,3,4,5,6,7,8,9] (can be measured experimentally using known probes or can be calculated based on the parameters of the image recording device), and with the exact PSF better results can be provided using less computational power. However, in less fortunate cases, e.g. in astronomy and remote sensing, we might not know the PSF a priori, hence both the original image and the PSF has to be estimated "blindly" during the deconvolution process [10,11,12,13,14].

Deconvolution methods can also be classified to iterative [2,3,4,5,6,12,13] or non-iterative [7,8,9] algorithms. Iterative methods provide an estimation of the original image and - in case of blind deconvolution - the PSF in each iteration, while non-iterative methods work as filters on the measured image. Non-iterative methods are usually simple and fast, but they amplify noise and suppress high frequency components in the restored image, while iterative methods offer a solution for these problems at the cost of more computational power and ill-posedness of the estimated results [1]. Detailed information about classification of deconvolution methods can be found in [1].

In this article we will focus on a common issue of iterative methods, the stopping condition. In the following experiments we use an iterative, non-blind deconvolution algorithm, described in [5,6], and we will describe the proposed method for the non-blind case.

1.1 The Necessity of Stopping Condition for Iterative Methods

Since we do not know the original image (U), only the blurry measured one (Y) can be used to guide us toward U . If $X(t)$ is the output of the method after t iteration, then a goal function of the method is usually based on minimizing the following expression:

$$\min_t (|H * X(t) - Y|) \quad (2)$$

Obviously the goal is to minimize $|U - X(t)|$, by stopping the iterations at that point where the Mean Square Error (*MSE*) is minimal. However, we can only access $|Y - H * X(t)|$. Let $X(t)$ be an iterated estimation, while another one is $X'(t) = X(t) + N(t)$, where they differ in an additional $N(t)$ noise and residual error with zero mean. In this case $H * N(t) \approx 0$, and $Y'(t) = H * X'(t) = H * X(t) + H * N(t) \approx H * X(t)$. Since the iterations are controlled by $H * X(t)$, this allows possible cases for $t_n \neq t_m$ where $|X'(t_n) - U| \gg |X(t_m) - U|$ is true, while $|H * X'(t_n) - Y| \leq |H * X(t_m) - Y|$ (see Fig. 3), and this is why the problem is ill-posed. As stated in [15], this problem affects the quality of solution of the iterative algorithms highly.

1.2 Existing Stopping Conditions

One way to try to stop this corruption is making additional assumptions about the target image (like a non negativity constraint, or smoothness constraint). This is called regularization of the deconvolution methods [16,17]. Regularization can enhance the image quality but the output highly depends on the chosen

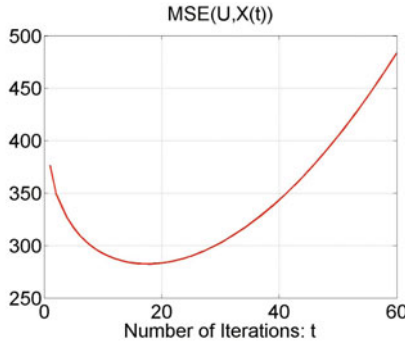


Fig. 1. As the number of iterations t passes the ideal stopping point, the $MSE(U - X(t))$ starts to increase

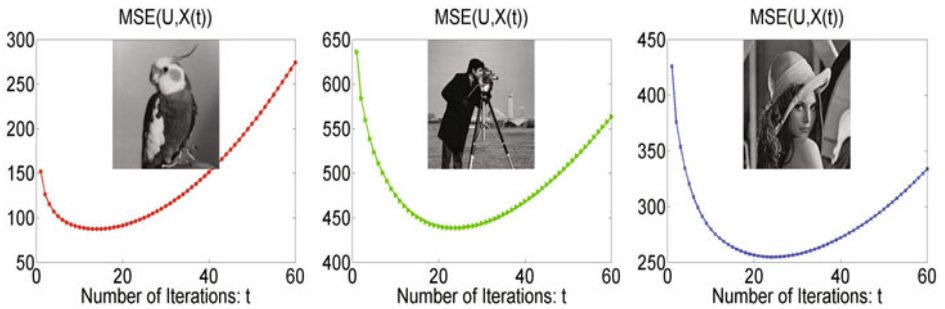


Fig. 2. Examples for the restored image quality versus iteration number in case of different images at the same blurring/deblurring kernel and noise level: no optimal number of iterations can be tuned

regularization parameters. In the following we will show that optimal result can be achieved without regularization, based on the noise independence criterion only.

Another way is to estimate the number of iterations needed to reach the best image quality and stop the process before the image gets corrupted. A straightforward idea is to stop the process after a constant number of iterations. Based on our experiments, this constant is around 7-10 iterations for a Lucy-Richardson based method [2,3] like the one we used [5,6]. For other methods this constant is different (see [15]).

Another way is to stop the iteration after the change between two consecutive estimations of the image becomes lower than a certain threshold [18]. In the following we will call this Differential Based Stopping Condition (DBSC):

$$DBSC : \frac{|X(t) - X(t - 1)|}{|X(t)|} < th \tag{3}$$

where th is a heuristic choice for threshold, usually between 10^{-3} and 10^{-6} . We have also tested a modified version of the above condition (in the following: MDBSC), where the re-blurred estimated images ($H * X(t)$) were considered:

$$MDBSC : \quad \frac{|H * X(t) - H * X(t-1)|}{|H * X(t)|} < th \quad (4)$$

Other similar solutions are summarized in [15]. The problem with these methods is that the number of iterations needed to reach the optimal restored image depends on many things: the picture itself, the PSF and the additional noise. See Fig. 2.

2 Orthogonality Based Stopping Condition

Rephrasing the problem, we can say that we are searching for the minimum of the MSE of $U - X(t)$:

$$\min_t (MSE(U - X(t))) \quad (5)$$

The problem is that we do not know U , hence the above function cannot be calculated. We need another function that expresses a similar thing but uses only known images.

In recent years a new estimation error has been introduced for similar purposes. Its efficiency has been proved for focus measurement in blind deconvolution problems, see [19]. This error definition, called Angle Deviation Error (ADE), is based on the orthogonality principle [20], considering the independence of noise and the estimated signal:

$$ADE(Q, P) = \left| \arcsin \left(\frac{\langle Q, P \rangle}{|Q| \cdot |P|} \right) \right| \quad (6)$$

We will show that conventional measures, like MSE, cannot help us to find optimal stopping criteria; while ADE has an optimum, close to the minimum of $|X - U|$. In our case the problem with MSE is that it measures similarity between two images. Since U is unknown, we cannot calculate $MSE(U - X(t))$, the only remaining logical possibility is to use Y and $H * X(t)$. These two values are the base of the goal function of the iterative deblurring algorithms, which leads us back to the original ill-posedness problem, meaning that the $MSE(Y - H * X(t))$ will decrease monotonously, although after a while high frequency noise will appear in $X(t)$. See Fig. 3.

In general we cannot measure the noise, and we only have data about the measurement Y . However, when estimating the $X(t)$ image, we can say that when further iterations would not enhance the image, then the error between two consecutive estimated images ($X_e(t)$) is independent of the real error ($X_{re}(t)$):

$$X_e(t) = X(t) - X(t-1) \quad (7)$$

$$X_{re}(t) = X(t) - U \quad (8)$$

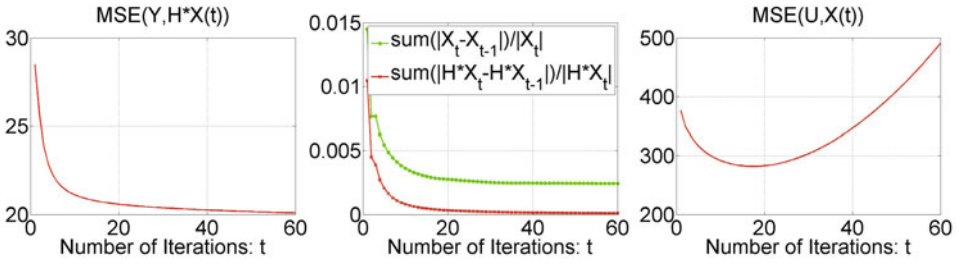


Fig. 3. The measurable function $MSE(Y - H * X(t))$ (on the left) or the difference functions in DBSC, MDBSC (in the middle) do not follow the unknown function $MSE(U - X(t))$ (on the right)

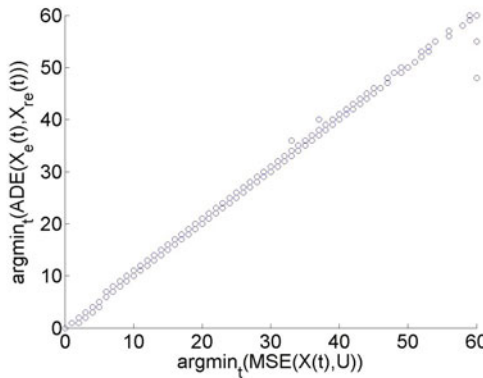


Fig. 4. The relationship between the minimum of MSE and $ADE(X_e, X_{re})$ for various pictures with different SNR and blur radii

Otherwise we should continue the iteration to decrease the error. Once the independence has been reached, the process must be stopped, since any steps after this point may fall into the doubtful region of ill-posedness. The best optimization could be given by the independence of $X(t) - X(t - 1)$ and $X(t) - U$, in other words we have to stop the iterative process when the following ADE function reaches its minimum:

$$ADE(X_e, X_{re}) = \left| \arcsin \left(\frac{\langle X_e, X_{re} \rangle}{|X_e| \cdot |X_{re}|} \right) \right| \tag{9}$$

It is clearly shown, as Fig. 4 plots, that the minimum location of $ADE(X_e, X_{re})$ correlates well with the minimum of $MSE(U - X(t))$; however, $ADE(X_e, X_{re})$ still contains the unknown image U . Another option is to use the $Y - Y(t)$ instead of the unknown X_{re} , but it is disturbed by the blurring function H in $Y(t)$. The clearest way for capturing the independence of the signal and the noise is using the error between two consecutive estimated images $X_e(t)$ and the unblurred estimation $X(t)$:

$$ADE(X_e, X(t)) = \left| \arcsin \left(\frac{\langle X_e, X(t) \rangle}{|X_e| \cdot |X(t)|} \right) \right| \tag{10}$$

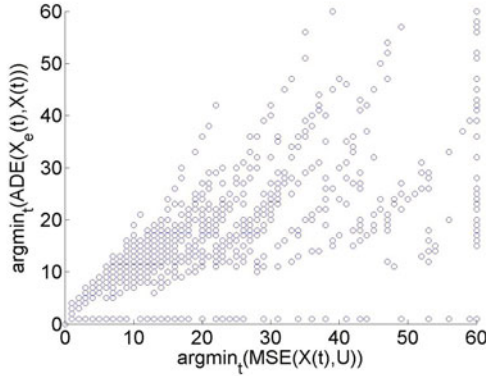


Fig. 5. The relationship between the minimum of $MSE(U - X(t))$ and $ADE(X_e, X(t))$ for various pictures with different SNR and blur radii

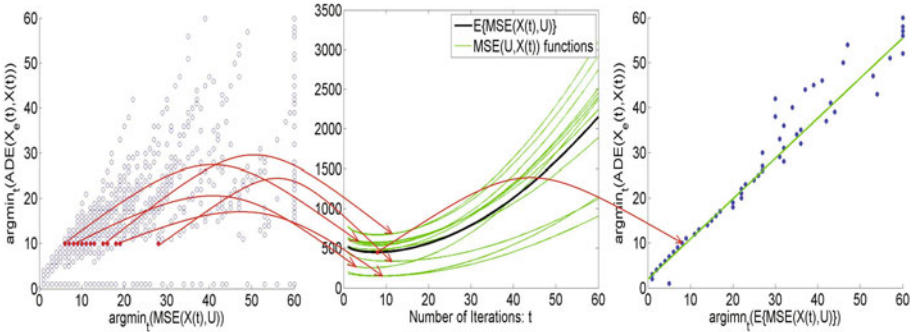


Fig. 6. The figure illustrates the calculation of a best common stopping point (*bcs*)

The above expression contains only measurable images and provides a reasonable solution for the stopping problem. At the minimum of $ADE(X_e, X(t))$ the change between two consecutive iterations (X_e) has the highest possible independence of the actual reconstructed image, hence we can assume that at this point X_e contains mostly independent noise and not structural information of the image, and further iteration will not enhance the image quality. In the following we will show how close this approximation brings us to the optimal stopping point.

In an ideal case, when the remaining $X_e(t_n)$ contains only random noise and for a given t_n the scalar product of $X_e(t_n)$ and $X(t_n) - U$ of Eq. (9) is zero (which means that the iterated change is independent of the structural differences of the restored image):

$$\langle X_e(t_n), X(t_n) - U \rangle = 0 \tag{11}$$

Then, using the distributive property of scalar product:

$$\langle X_e(t_n), X(t_n) - U \rangle = \langle X_e(t_n), X(t_n) \rangle - \langle X_e(t_n), U \rangle \tag{12}$$

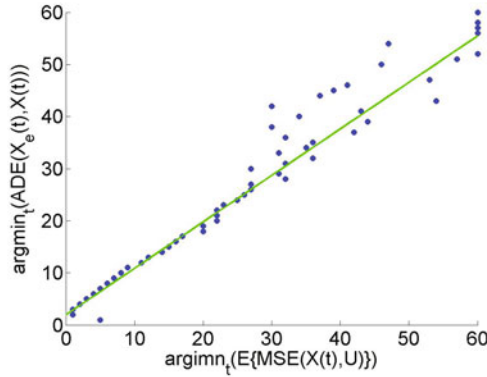


Fig. 7. The figure shows the calculated *bcspl* locations versus the proposed stopping points, $\alpha = \arg \min_t(ADE(X_e, X(t)))$

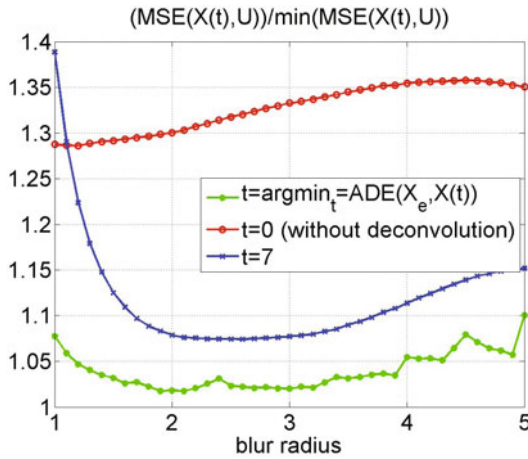


Fig. 8. The quality of the deconvolved image using a fixed iteration number and the proposed method as stopping condition. The quality of the blurred image is also shown for unchanged ($t = 0$) images as reference.

Since U may contain high frequency components correlating with X_e , the possible zeros of components in Eq. (12), $\langle X_e(t_n), X(t_n) - U \rangle = 0$ and $\langle X_e(t_m), X(t_m) - U \rangle = 0$, are not necessarily coincident, since $\langle X_e(t), U \rangle \neq 0$ biases Eq. (12): $t_m \neq t_n$. We may say that, even for the most ideal case, the ill-posed property of the problem results in the biasing of the possible minimum, $t_n \neq t_m$. Fig. 5 shows the relationship between the minimum locations of $ADE(X_e, X(t))$ and $MSE(U - X(t))$ functions. Although, due to the above reasons the correlation is not as perfect as it was between $ADE(X_e, X_{re})$ and $MSE(U - X(t))$, it is clearly visible.

However, what really matters is not the difference in the number of executed iterations, but the difference between the MSE values at the two functions'

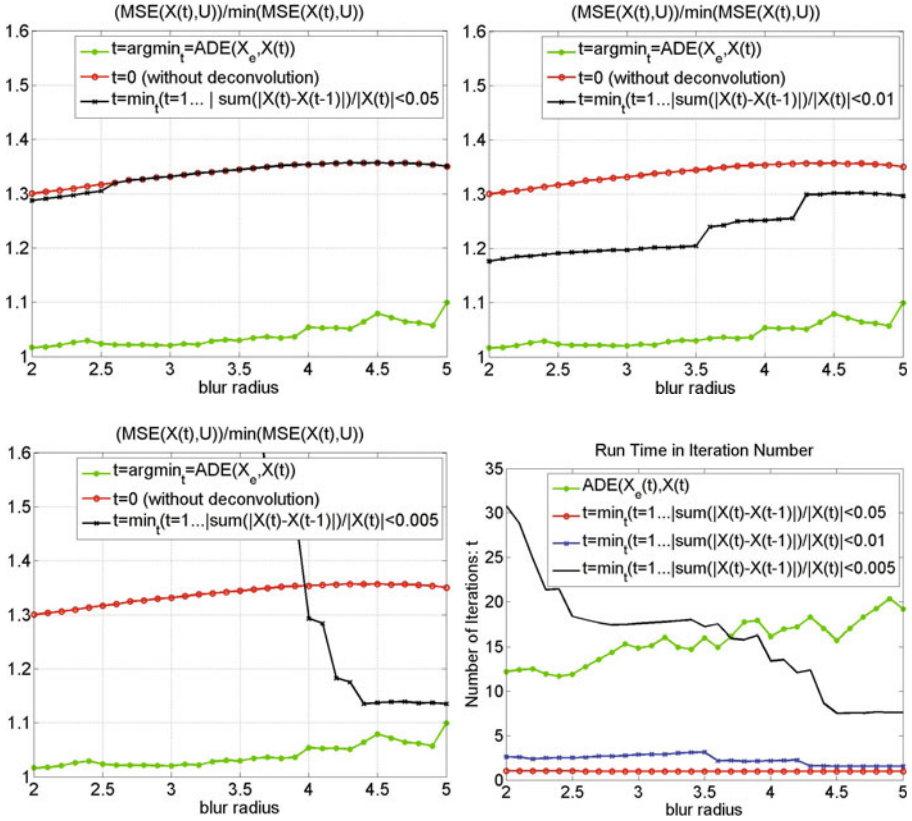


Fig. 9. The quality of the deconvolved image using DBSC (Eq. (3)) with different thresholds and the proposed method, and the runtime of the methods in number of iterations. The quality of the blurred image is also shown for unchanged ($t = 0$) images as reference.

minimum location. It is possible that the MSE function has a long quasi constant part around the minimum, see the mid image of Fig. 6. In this situation the difference between the executed iterations is high, while the value of the MSE function practically does not change. Our main priority is to provide an estimation $X(\alpha)$ that is as close to the best possible iteration $X(\beta)$ as possible, where $\beta = \operatorname{argmin}_t(MSE(U - X(t)))$ and $\alpha = \operatorname{argmin}_t(ADE(X_e, X(t)))$. The shape of $MSE(U - X(t))$ function changes from image to image, and the blur radius or the SNR of Y may also affect it, therefore the distance between β and α is not the best measure for us. Fig. 5 shows the relation between β and α for different measurements (each corresponds to a point on the figure).

There are points whose $\alpha = \operatorname{argmin}_t(ADE(X_e, X(t)))$ are the same (they are in the same "line" on Fig. 5). These measurements were stopped at the same α point by our stopping condition, but the corresponding β locations can be different.

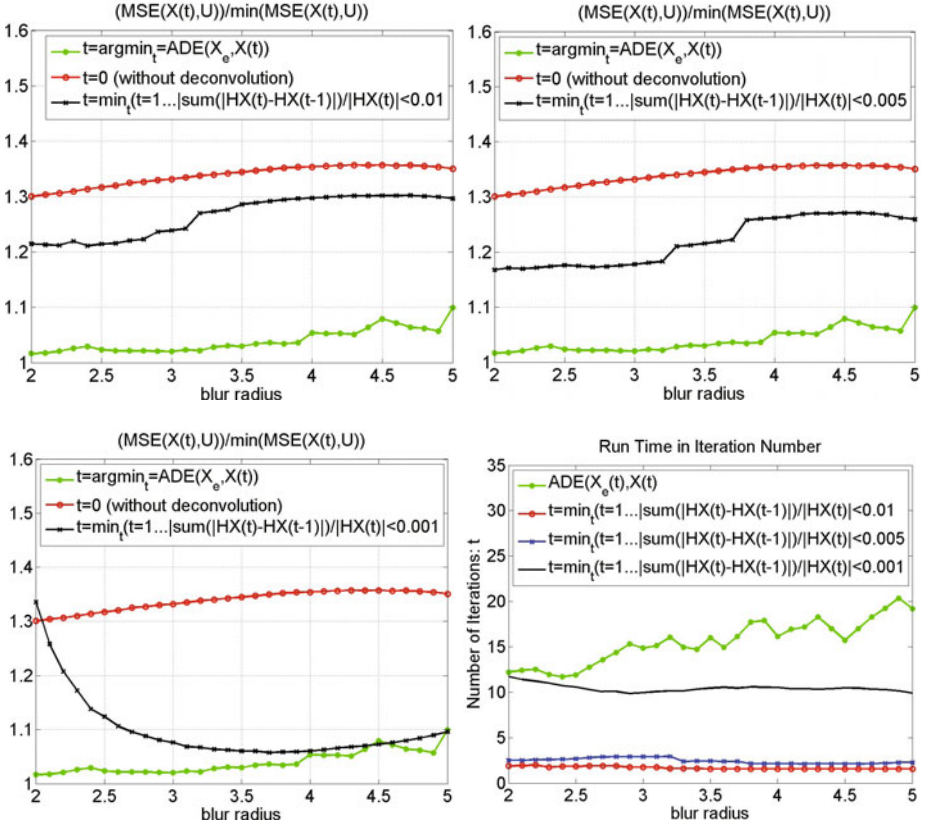


Fig. 10. The quality of the deconvolved image using MDBSC (Eq. (4)) with different thresholds (0.01, 0.005, 0.001) and the proposed method, and the runtime of the methods in number of iterations

This means that α might not be the best stopping point for each measurement individually, but we will show that considering all the measurements with the same α , it is close to the best common stopping point (*bcs**p*). To find this *bcs**p* for these cases, an average of their $MSE(U - X(t))$ functions has been calculated: $E\{MSE(U - X(t))\}$. The *bcs**p* will be the minimum location of $E\{MSE(U - X(t))\}$:

$$bcs\ p = \arg \min_t (E\{MSE(U - X(t))\}) \quad (13)$$

The calculated *bcs**p* locations can be seen versus the proposed stopping points, $\alpha = \arg \min_t (ADE(X_e, X(t)))$ in Fig. 7. We can conclude that, although our estimation is not always correct, for the measurements with the same α the expected value of β is close to α . An illustration of the calculation of a *bcs**p* point is shown in Fig. 6.

3 Results

In this section we will present the test conditions and the results achieved with the proposed algorithm and other competing methods. We used 25 images as database, which contain landscapes, images of buildings, animals, textures, black and white drawings. The PSF is a Gaussian kernel defined by different blur radii between 1 and 5. To the blurred images Additional White Gaussian Noise (AWGN) was added with SNR=20, 25, 30, 35, 40dB.

3.1 Comparative Results

To compare the proposed method to other existing stopping conditions, we calculated the ratio between the MSE value at the real minimum location (see Fig. 11) and at the point where the stopping condition would stop the iteration. We compared the proposed method to fixed iteration count (using the best possible fixed number in the test), the DBSC (Eq. (3)), the MDBSC (Eq. (4)) and as a baseline we also calculated the above mentioned ratio for the blurred image, Y . The results can be seen on Figs. 8, 9 and 10.

Fig. 8 compares the proposed method with a commonly used solution, where the deconvolution process is stopped after a constant number of iterations. In our experiments the best results were obtained when this constant was 7. The experiments were taken using all the 25 images with different blur radii and noise levels.

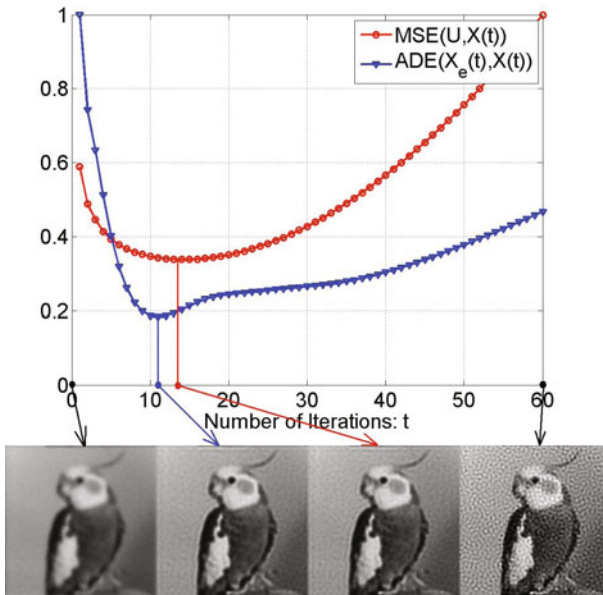


Fig. 11. The estimation results by using measurable $ADE(X_e, X)$ and non-measurable $MSE(U, X)$ functions at different iterations

Our tests demonstrate that the commonly used X_e based (DBSC) stopping condition is outperformed by the one using the blurred comparison (MDBSC). At any ad-hoc threshold settings, both of them are highly outperformed by the proposed ADE based function (see Fig. 9 and 10).

Fig. 11 shows estimation of the original image, U at different t points of the iteration process.

The results show that our ADE stopping criterion gives the best SNR estimation of U along with a well balanced run-time effort.

4 Conclusion

In the paper a novel method has been introduced for calculating the ideal stopping point for iterative non-regularized deconvolution processes. The proposed method is capable of estimating the optimal stopping point of iterations based on the independence of an actual estimated signal and its measurement controlled gradient, indicating when an aimless section of the iterations is just starting. It is stable in terms of quality and runtime, and it clearly outperforms the generally used ad-hoc methods. Although the results were obtained with a non-blind method, we did not apply any constraints about dimensionality or regularization issues. The same method might be well applied on double (blind 14) deconvolution as well: Eqs. (2) - (10) will be kept unchanged even in these cases. This is a possible direction for future work.

Acknowledgments. This work was partially supported by the Hungarian Research Fund, OTKA No. 105.719.

References

1. Sarder, P., Nehorai, A.: Deconvolution methods for 3-d fluorescence microscopy images. IEEE In Signal Processing Magazine 23, 32–45 (2006)
2. Lucy, L.: An iterative technique for rectification of observed distributions. The Astronomical Journal 79, 745–765 (1974)
3. Richardson, W.: Bayesian-based iterative method of image restoration. JOSA 62, 55–59 (1972)
4. Agard, D.: Optical sectioning microscopy: Cellular architecture in three dimensions. Ann. Rev. Biophys. Bioeng. 13, 191–219 (1984)
5. Biggs, D.S.C., Andrews, M.: Acceleration of iterative image restoration algorithms. Appl. Opt. 36, 1766–1775 (1997)
6. Hanisch, R.J., White, R., Gilliland, R.: Deconvolutions of hubble space telescope images and spectra. In: Jansson, P.A. (ed.) Deconvolution of Images and Spectra, 2nd edn., Academic Press, CA (1997)
7. Erhardt, A., Zinser, G., Komitowski, D., Bille, J.: Reconstructing 3-d light-microscopic images by digital image processing. Appl. Opt. 24, 194–200 (1985)
8. McNally, J.: Three-dimensional imaging by deconvolution microscopy. Methods 19, 373–385 (1999)
9. Tikhonov, A.N., Arsenin, V.Y.: Solutions of ill-posed problems. Scripta series in mathematics, Winston, Washington (1977)

10. Ayers, G.R., Dainty, J.C.: Iterative blind deconvolution method and its applications. *Opt. Lett.* 13, 547–549 (1988)
11. Markham, J., Conchello, J.A.: Parametric blind deconvolution: a robust method for the simultaneous estimation of image and blur. *J. Opt. Soc. Am. A* 16, 2377–2391 (1999)
12. Pankajakshan, P., Zhang, B., Blanc-Féraud, L., Kam, Z., Olivo-Marin, J.C., Zerubia, J.: Blind deconvolution for thin-layered confocal imaging. *Appl. Opt.* 48, 4437–4448 (2009)
13. Jang, K.E., Ye, J.C.: Single channel blind image deconvolution from radially symmetric blur kernels. *Opt. Express* 15, 3791–3803 (2007)
14. Kundur, D., Hatzinakos, D.: Blind image deconvolution. *IEEE Signal Processing Magazine* 13, 43–64 (1996)
15. Verbeeck, J., Bertoni, G.: Deconvolution of core electron energy loss spectra. *Ultramicroscopy* 109, 1343–1352 (2009)
16. Dey, N., Blanc-Fraud, L., Zimmer, C., Kam, Z., Roux, P., Olivo-Marin, J., Zerubia, J.: Richardson-lucy algorithm with total variation regularization for 3d confocal microscope deconvolution. *Microscopy Research Technique* 69, 260–266 (2006)
17. van Kempen, G., van Vliet, L.: The influence of the regularization parameter and the first estimate on the performance of tikhonov regularized nonlinear image restoration algorithms. *J. Microsc.* 198, 63–75 (2000)
18. You-Wei Wen, A.M.Y.: Adaptive parameter selection for total variation image deconvolution. *Numer. Math. Theor. Meth. Appl.* 2, 427–438 (2009)
19. Kovács, L., Szirányi, T.: Focus area extraction by blind deconvolution for defining regions of interest. *IEEE Tr. Pattern Analysis and Machine Intelligence* 29, 1080–1085 (2007)
20. Papoulis, A.: *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, New York (1984)

Probabilistic 3D Object Recognition Based on Multiple Interpretations Generation

Zhaojin Lu¹, Sukhan Lee^{1,*}, and Hyunwoo Kim²

¹ Intelligent System Research Center, School of Information and Communication Engineering, SungKyunKwan University, Suwon, Korea

luzhaojin@skku.edu, lsh@ece.skku.ac.kr

² Department of New Media, Korean German Institute of Technology, Seoul, Korea

hwkim@kgit.ac.kr

Abstract. We present a probabilistic 3D object recognition approach using multiple interpretations generation in cluttered domestic environment. How to handle pose ambiguity and uncertainty is the main challenge in most recognition systems. In our approach, invariant 3D lines are employed to generate the pose hypotheses as multiple interpretations, especially ambiguity from partial occlusion and fragment of 3D lines are taken into account. And the estimated pose is represented as a region instead of a point in pose space by considering the measurement uncertainties. Then, probability of each interpretation is computed reliably using Bayesian principle in terms of both likelihood and unlikelihood. Finally, fusion strategy is applied to a set of top ranked interpretations, which are further verified and refined to make more accurate pose estimation in real time. The experimental results support the potential of the proposed approach in the real cluttered domestic environment.

1 Introduction

3D object recognition and pose estimation is a difficult problem in computer vision and intensively investigated for many years with widespread applications. How to deal with ambiguity and uncertainty of 3D object recognition in domestic environment with change of illumination, perspective viewpoint, distance, partial occlusion and background, etc is still an open problem.

Many researchers proposed various 3D object recognition approaches since Robert's pioneer work [1]. Among them, model-based recognition method is most general one, which computing the hypothesized model pose by finding correspondences between the model features and image features, and final pose is verified with additional image features. Fischler and Bolles' RANSAC approach [2], Beis and Lowe's invariants indexing approach [3], and Costa's relational indexing approach [4]. These approaches, which hypothesize poses from initial feature matching correspondences and verify those hypotheses based on additional presence of supporting correspondences, cannot be in real time when the number

* Corresponding author.

of model and image features becomes large. David et al. [5] proposed the approach that the recognition and pose estimation are solved simultaneously by minimizing energy function. But it may not be converged to minimum value in functional minimization method due to high non-linearity of the cost function.

Recently, there have been a number of appearance based approaches to 3D object recognition in which multiple 2D views are sampled as the representation of 3D objects [6,7]. Cyr and Kimia [8] employed an aspect-graph view-based method, where the viewing sphere is sampled at regular (five-degrees) intervals and the similarity metric is used in an iterative procedure to combine views into aspects, with a prototype representing each aspect. Sun [9] proposed a multi-view probabilistic model, which considerate not only similar features in multi-view images, but also 3D relationship among the multi-views or multiple parts of one view. However, these methods cannot provide accurate pose estimation since they do not use 3D models, and are sensitive to illumination change, clutter and partial occlusion.

More recently, the use of range images has been popular as a way of overcoming the limitation of 2D images. In range images, 3D shape are represented by local feature, such as spin images [10], 3D shape context [11] [12] are examples where surface points are described by shape distribution of a local neighborhood. However, these methods mostly deal with dense and accurate depth data, which is suitable for stereo vision based images.

Shimshoni and Ponce [13] proposed a probabilistic approach for 3D object recognition, which used 2D line features and 2D models sampled from viewing sphere using probabilistic peaking effect, but they assume that the lines in 3D object can be covered well in image by edge/line detector, and the computation is so expensive. David's [14] also proposed a 2D line feature based approach for 3D object recognition, which only can recognize object within 30° away from the modeled viewpoints, which means a large number of models should be made for each object.

In domestic environment, most of the objects are rigid with straight line features, such as table, refrigerator, book, TV, milk box, etc. Most of the information available for object recognition is 3D lines because 3D data can be obtained robustly with stereo camera on the boundaries of objects even in texture-less objects. Approaches to object recognition that rely on 2D features only are likely to perform poorly because 2D models are normally viewpoint-dependent. Therefore, in order to recognize the 3D object from any viewpoint and distance for service robot. To deal with this challenge, invariant 3D features should be adopted. This paper presents an effective probabilistic method for recognizing 3D object in domestic environment for home service robot, where both the object models and their images are represented by invariant geometric features, 3D line segments.

There are three challenges in constructing such a 3D object recognition system. The first is how to generate multiple interpretations to cover all the possible locations of the target object in 3D space. In our approach, we group the 3D lines into two types of feature sets, pairs of parallel lines and pairs of perpendicular

lines, which usually appear in man-made objects. Every pairing of image feature set to a model feature set contributes a pose hypothesis as an interpretation, typically, each image feature set corresponding to several model feature sets, which results in multiple interpretations. Thanks to the adopted invariant 3D line feature, compare to conventional 2D feature based approaches, the total number of interpretations is much fewer, and each interpretation is less likely to be corrupted by spurious features.

The second challenge is how to verify each interpretation with additional evidences, most of the initial hypothesized interpretations are inaccurate because correspondences between the model feature sets and image feature sets are incorrect. Thus, our approach ranks interpretations in a probabilistic manner using Bayesian rule. To ensure the estimated probability is reliable, the proposed method estimates the probability not only through likelihood measurement between the model feature sets and image feature sets, but also through unlikelihood measurement by analyzing the spatial distribution of the support evidences. The new probability estimation is largely robust to environment change. In order to take into account the uncertainty in the values measured in the image, we represent each interpretation as a region in the pose space rather than a point in that space. Consequently, each interpretation is represented as a Gaussian *pdf* with a certain probability weight.

The final challenge is how to refine top ranked interpretations to provide more accurate pose, we make use of the information inherent in interpretations, which means interpretations should yield compatible poses if they correspond to the same object. We fuse sets of interpretations which support each other and output a small number of fused interpretations with higher probabilities and smaller uncertainties. Compare to modified Gold's graduated assignment algorithm [14], which needs a number of iterations using deterministic annealing to get optimal pose. By fusing the compatible set of interpretations, we are able to find a correct precise pose in real time.

The remainder of the paper is structured as follows: Detail of multiple interpretations generation is described in section 2. In section 3 we derive probability computation in terms of likelihood and unlikelihood. In section 4 we present fusion based pose refinement. Section 5 demonstrates experimental results, and the conclusions are given in Section 6.

2 Multiple Interpretations Generation

The motivation of probabilistic multiple interpretations specifically focus on 3D object recognition in domestic environment, the feature selected as weak evidence for the initial object recognition is incomplete and ambiguous, meaning that the feature may generate a number of matching that are not the target object we are searching for or, even though the feature represents the target object, the feature is not able to localize the target object uniquely as Fig. 1 shows. Furthermore, we need to consider the case where maybe multiple of similar objects present in the scene. How to incorporate the above two factors into object recognition is a matter

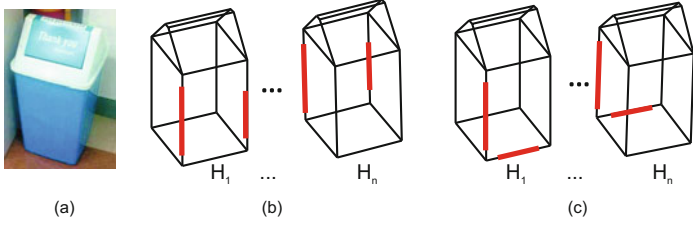


Fig. 1. Multiple interpretations. (a) target object, (b) parallel line pairs based interpretations, (c) perpendicular line pairs based interpretations. Where H_1, \dots, H_n represent hypotheses from different feature sets of model.

of interest. As stated above, the initial matching generated by the initial feature are rather ambiguous and incomplete in terms of uniquely identifying where the target object is. In order to generate the true interpretations, each matching is interpreted in terms of possible object poses. These newly identified interpretations are now subject to further evaluation with additional evidences so as to determine the probability and that the candidate represents the target object.

Parallel line pairs and perpendicular line pairs are typical combination of line features of human made object in domestic environment as Fig 1 shows. We would like to compute $P(x, H_m, O|F)$, which is obtained basically from placing the model of target object O at location x given feature F , where H_m denotes hypothesis that matching m^{th} model feature of model O against the measured image feature set F . More specifically, $P(x, H_m, O|F)$ can be represented, from Bayesian principle, as

$$P(x, H_m, O|F) = P(H_m, O|F)P(x|F, H_m, O) \tag{1}$$

where $P(H_m, O|F)$ is the probability (i.e., a positive real value ≤ 1) that F represents hypothesis H_m of target object O . On the other hand, $P(x|F, H_m, O)$ represents the probability that hypothesis H_m of the object O , given F , is located at x . Since x represents a variable due to the uncertainty of image features, thus, $P(x|F, H_m, O)$ defines a probability density function(pdf) along x . Therefore, the pose of an interpretation is represented as a region instead of a point in the pose space. Due to the partial occlusion and fragmentation of 3D lines, the image feature set F may not uniquely match with the hypothesis H_m of target object O . We could use the midpoint to make the correspondence between each image line and model line, but this is not always true in the case of a short line superposed on a longer line (think of the short line is a fragment of the longer line). A solution to this challenge is to generate multiple representations for each image feature set, which means F can be represented by a series of extended F_k (as Fig 2 shows) that uniformly sample the dynamic range with an interval s , where the dynamic range is computed based on the length difference between F and H_m , and interval s depends on the size of model and dynamic range. Incorporating F_k into (1) yields:

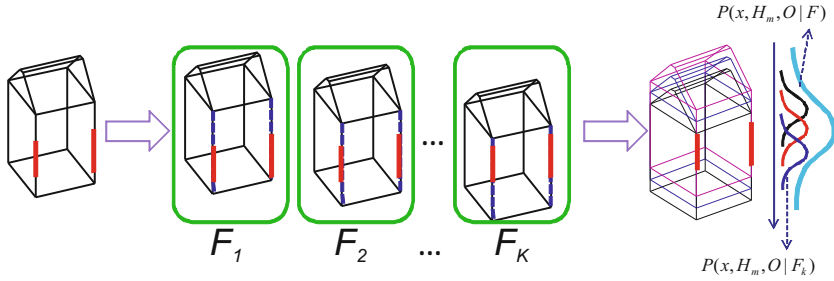


Fig. 2. Illustration of sub-interpretations, where F_1, F_2, \dots, F_m are extended features (represented as dashed blue line)

$$P(x, H_m, O|F) = \sum_{k=1}^K P(x, H_m, O|F_k)P(F_k|F) \quad (2)$$

where $P(F_k|F)$ is the probability that F_k represents F is correct, which is uniformly distributed as mentioned above. And $P(x, H_m, O|F_k)$ can be represented as $P(H_m, O|F_k)P(x|F_k, H_m, O)$ similar to the derivation in (1), where $P(x, H_m, O|F_k)$ is noted as sub-interpretation by matching the extended image feature set F_k against m^{th} model feature set of hypothesis H_m . We approximate $P(x|F_k, H_m, O)$ as a Gaussian *pdf* based on the range resolution of the stereo camera [15]. Therefore, $P(x, H_m, O|F)$ is computed by summing over the sub-interpretations, which is a mixture of Gaussians as Fig. 2 illustrates. Although the complexity increases linearly to the number of sub-interpretations, the recognition performance is greatly enhanced. Finally, the pose hypothesis of a sub-interpretation is generated as follows, in order to compute the pose, endpoints of each 3D line in H_m and F_k are corresponded one by one, totally we have four pairs of corresponding endpoints. Thus, the transformation mapping a model feature set H_m to the extended image feature set F_k is

$$F_k = T_m^k H_m \quad (3)$$

where T_m^k is a 4x4 homogeneous transformation matrix with twist representation [16], and the corresponding twist parameters are represented as a vector as $\theta_m^k = (\omega_x \ \omega_y \ \omega_z \ t_x \ t_y \ t_z)$, where $(\omega_x \ \omega_y \ \omega_z)$ represent rotation, and $(t_x \ t_y \ t_z)$ represent translation. For sake of explanation clearly in following sections, the general form of sub-interpretation is characterized as

$$I_m^k = \{\pi_m^k, N(\theta_m^k, \Sigma_m^k)\} \quad (4)$$

where π_m^k denotes the probability weight $P(H_m, O|F_k)$, and $N(\theta_m^k, \Sigma_m^k)$ denotes a Gaussian *pdf* of pose distribution $P(x|F_k, H_m, O)$.

3 Probability Computation

The second challenge of the proposed approach is that how to rank the generated multiple interpretations probabilistically. For this purpose, a matching

probability between the model (transformed by an estimated pose) and image is computed. Since each interpretation is represented by a number of sub-interpretations, so instead of computing the probability of each interpretation, we compute the probability of each individual sub-interpretation separately. In other words, rather than compute $P(H_m, O|F)$, we would like to compute $P(H_m, O|F_k)$, using Bayesian law

$$P(H_m, O|F_k) = \frac{P(F_k|H_m, O)P(H_m, O)}{P(F_k)} = \frac{1}{1 + \alpha} \quad (5)$$

where $\alpha = \frac{P(F_k|\overline{H_m, O})}{P(F_k|H_m, O)}$, and we assume $P(H_m, O)$ and $P(\overline{H_m, O})$ are equal, because no prior knowledge is available.

More specifically, $P(F_k|H_m, O)$ is the likelihood probability that feature F_k appears given hypothesis H_m of the object O is present in the scene, whereas $P(F_k|\overline{H_m, O})$ is the unlikelihood probability that feature F_k appears when the object O is absent in the scene. To ensure the estimated probability is reliable, all the neighboring 3D lines around the estimated pose should be involved as support evidences in probability computation. Let $\mathcal{N}(\theta_m^k)$ denotes all the neighboring 3D lines around estimated pose θ_m^k as shown in Fig. 3(a). So during the real computation of $P(H_m, O|F_k)$ in (5), F_k is alternatively represented as $\mathcal{N}(\theta_m^k)$.

Actually, the probability of each sub-interpretation is the function of the pose. More details about the definition of support evidence are shown in Fig. 3(b). Let L_j be j^{th} line segment of the sub-interpretation where $j \in [1, N_r]$, N_r is the number of the all visible line segments (solid black 3D lines in Fig. 3(a)) of the sub-interpretation. A line feature l_i^j belongs to L_j , determined by the distance from the mid-point of the line feature to L_j . The angle θ_i^j between the line feature l_i^j and L_j is also utilized. Two threshold values those are specified a priori are utilized to remove non-relevant line features such as \bar{d} and $\bar{\theta}$ for distance threshold and angle threshold, respectively. Due to line fragment, L_j might own several line features l_i^j , $i \in [1, N_j]$ where N_j is the number of line features that belong to L_j .

3.1 Likelihood Computation

In order to compute likelihood probability $P(F_k|H_m, O)$, we are opted to use not only error distance but also coverage of the feature line over the sub-interpretation. The error distance of the i^{th} line feature with respect to the j^{th} reference line segment is denoted by d_i^j and defined by the distance between from mid-point of the line feature to the reference line.

As mentioned above, each reference line might possess several line features within its threshold. In order to compute the coverage of each reference line, we projected each line feature onto the corresponding reference line. As shown in Fig. 3(c), the green portion of the reference line L_j represents the coverage of the line features with respect to the reference line. Subsequently, the error e_j and the coverage c_j associated with each j^{th} reference line are computed as

$$e_j = \min \left\{ E_{max}, \frac{1}{N_j} \sum_{i=1}^{N_j} \left(\mu \frac{d_i^j{}^2}{\bar{d}^2} + (1 - \mu) \frac{\tan^2 \theta_i^j}{\tan^2 \bar{\theta}} \right) \right\}, \quad c_j = \max \left\{ C_{min}, \sum_{i=1}^{N_j} \frac{l_i^j}{L_j} \right\} \quad (6)$$

where the parameter E_{max} and C_{min} ensure that “good” poses are not penalized too severely when a model line is fully occluded in the image. This parameter is easily set by observing the values of e_j and c_j that are generated for poor poses. It should be noted that when calculating each error e_j , distance and the angle error are normalized by the threshold values \bar{d} and $\bar{\theta}$. In particular, the coefficients μ is utilized to impose relative weight between the distance error and the angle error. Therefore, given a set of the reference lines, the total error is computed by taking the averages of each error and coverage as $e = \frac{1}{N_r} \sum_{j=1}^{N_r} e_j$ and $c = \frac{1}{N_r} \sum_{j=1}^{N_r} c_j$. Finally, the likelihood probability is computed by (7).

$$P(F_k | H_m, O) = c(1 - e^2) \quad (7)$$

Note that the likelihood is proportional to the coverage, while being parabolic to the error.

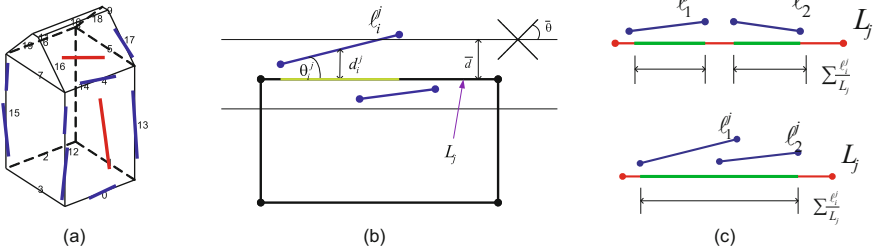


Fig. 3. Generated interpretation and support 3D line evidences, where (a) shows neighboring 3D measured lines which are belong to the estimated pose, blue 3D lines are neighboring but red lines are not, (b) shows the geometric constraints requirement of the support 3D line evidences, and (c) illustrates the coverage of support line features

3.2 Unlikelihood Computation

We define the unlikelihood as the probability of detecting a particular feature set under the absence of the target object. Most of the previous approaches compute the unlikelihood probability based on either learning or empirical data, where learning-based approach requires a large number of manually labeled training data, typically hundreds or thousands of images are required, and empirical data is obtained from some typical scenes, which suffers the problem of accuracy and robustness. Whereas we propose a novel approach for unlikelihood evaluation in a computational way in terms of diversity, which is defined by analyzing the spatial distribution of the support evidences. For instance, as shown

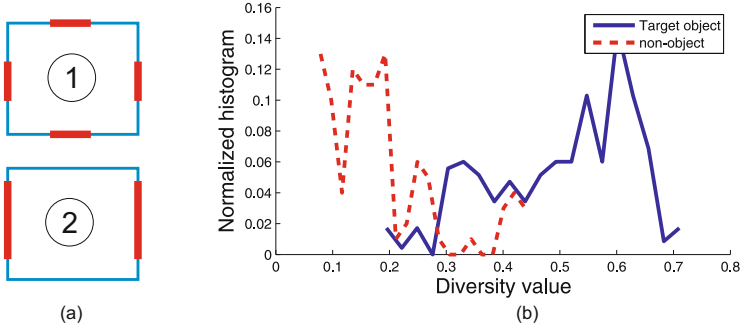


Fig. 4. (a) illustrates diversity and coverage of support line evidences, and (b) shows statistics distribution of diversity value for target object and non-object

in Fig. 4(a), where ① and ② have the same overall coverage (covered by thick red lines), but actually ① is more robust than ② because lines are equal distributed in ①, which provides stronger geometric constraints. Therefore, if the support evidences are distributed more equally around the estimated pose, then this estimated pose is more likely to be the true object, otherwise, the estimated pose might be generated from background or other non-objects with similar initial feature set. For this purpose, we compute the unlikelihood probability to evaluate whether the estimated pose is reasonable.

If the overall coverage is fixed, then diversity is maximized when c_j (defined in (6)) are totally equal. Under this criterion, the problem of diversity computation is equal to entropy computation, which is well defined in standard information theory. It follows that diversity maximization is equivalent to maximize entropy, which is also equivalent to infomax. The equivalence between diversity maximization and infomax can make sense, because if the support evidences around the estimated pose can provide most information of the target object, then the estimated pose is more likely to be correct. Remembering that the unlikelihood probability is defined under the assumption that object is absent in the scene. So if the object is absent or invisible in the scene, then the support evidences might not provide much related information about the target object.

The unlikelihood probability is computed as follows. Let \tilde{c}_j denote the normalized c_j as $\tilde{c}_j = \frac{c_j}{\sum_{s=1}^{N_r} c_s}$, so that $\sum_{j=1}^{N_r} \tilde{c}_j = 1$. Then, the diversity of the support evidences $\mathcal{N}(\theta_m^k)$ around the estimated pose θ_m^k is computed as

$$D(\mathcal{N}(\theta_m^k)) = \sum_{j=1}^{N_r} (-\tilde{c}_j \cdot \log(\tilde{c}_j)) \tag{8}$$

In order to represent the unlikelihood probability in terms of diversity $D(\mathcal{N}(\theta_m^k))$, we need to normalize the value of $D(\mathcal{N}(\theta_m^k))$ in the range of $[0, 1]$. Since the max value of $D(\mathcal{N}(\theta_m^k))$ is $\max\{D(\mathcal{N}(\theta_m^k))\} = \sum_{j=1}^{N_r} \left(-\frac{1}{N_r} \cdot \log\left(\frac{1}{N_r}\right)\right) \equiv \log(N_r)$.

Hence, the normalized diversity $\tilde{D}(\mathcal{N}(\theta_m^k))$ is represented as

$$\tilde{D}(\mathcal{N}(\theta_m^k)) = \frac{D(\mathcal{N}(\theta_m^k))}{\log(N_r)} \quad (9)$$

Finally, the unlikelihood probability is computed as

$$P(F_k | \overline{H_m, O}) = 1 - \tilde{D}(\mathcal{N}(\theta_m^k)) \quad (10)$$

The smaller the value of $P(F_k | \overline{H_m, O})$, the more likely the estimated pose θ_m^k is correct. In order to justify the effectiveness of diversity computation, we captured a number of images for testing, half of them contain target object, and another half contain non-object, followed by multiple interpretations generation. Statistic distribution of diversity value is shown in Fig 4(b). From this we can see that, the diversity value of non-object and target object can be discriminated well.

4 Pose Verification and Refinement

The final challenge of the proposed approach is to apply a pose refinement to a few top ranked interpretations according to the estimated probabilities. Remembering that we represent an interpretation as a number of sub-interpretations during the stage of multiple interpretations generation. The main purpose of the sub-interpretations is that decrease the ambiguity due to the partial occlusion or fragment of the 3D line features. Since after probability computation stage, we know the relative importance among sub-interpretations, and these sub-interpretations have **OR** relationship which means only one sub-interpretation is correct in an interpretation. Therefore, in the refinement stage, we only choose one sub-interpretation with highest probability to represent an interpretation. In other words, an interpretation is represented as a weighted Gaussian *pdf* in this stage.

For the sake of simplicity, let's assume that two independent image feature sets f_1 and f_2 and two respective hypotheses h_1 and h_2 are given, which result in two interpretations as $P(x, h_1, O | f_1)$ and $P(x, h_2, O | f_2)$, and their corresponding general form representations are given as $I_1 = \{\pi_1, N(\theta_1, \Sigma_1)\}$ and $I_2 = \{\pi_2, N(\theta_2, \Sigma_2)\}$ respectively as defined in (4). Therefore, The fusion of two interpretations is performed by fusing two weighted Gaussian *pdfs*. Now, we are interested in the possibility of fusing the two interpretations and, if possible, how to fuse them into $P(x, h_1, h_2, O | f_1, f_2)$, or how to get its corresponding general form $I = \{\pi, N(\theta, \Sigma)\}$.

For f_1 and f_2 to be features of the same instance of object, the pose of the object x must be in the intersection of the pose uncertainty regions of two interpretations. But if f_1 and f_2 do not belong to the same instance object, then they do not support each other. Therefore, before fusion two interpretations, we need to check the support relationship between them. For this purpose, the support is determined by Mahalanobis distance between two pose distributions, $N(\theta_1, \Sigma_1)$ and $N(\theta_2, \Sigma_2)$, which is computed as

$$d(I_1, I_2) = \sqrt{(\theta_1 - \theta_2)^T \left(\frac{\Sigma_1 + \Sigma_2}{2} \right)^{-1} (\theta_1 - \theta_2)} \tag{11}$$

if $d(I_1, I_2) \leq \delta_{th}$, then they are deemed to support each other. We have found that when $\delta_{th} = 3$ can achieve good performance. And then how to fuse two interpretations in terms of probabilities and pose distributions becomes a matter of interest. Note that $P(x, h_1, h_2, O|f_1, f_2) = P(h_1, h_2, O|f_1, f_2)P(x|f_1, f_2, h_1, h_2, O)$. Similar to (5), fused probability $P(h_1, h_2, O|f_1, f_2)$ is computed as

$$P(h_1, h_2, O|f_1, f_2) = \frac{P(f_1, f_2|h_1, h_2, O)}{P(f_1, f_2)} = \frac{1}{1 + \alpha_{12}} \tag{12}$$

where $\alpha_{12} = \frac{P(f_1|h_1, O)}{P(f_1|h_1, O)} \cdot \frac{P(f_2|h_2, O)}{P(f_2|h_2, O)}$. In addition to probability fusion, pose distribution fusion is computed as

$$P(x|f_1, f_2, h_1, h_2, O) = P(x|f_1, h_1, O)P(x|f_2, h_2, O) \tag{13}$$

Since both $P(x|f_1, h_1, O)$ and $P(x|f_2, h_2, O)$ have their corresponding weights, To adapt to reliability of each interpretation, so (13) is slightly changed as

$$P(x|f_1, f_2, h_1, h_2, O) = P(x|f_1, h_1, O)^{\omega_1} P(x|f_2, h_2, O)^{\omega_2} \tag{14}$$

where $\omega_1 = \frac{\pi_1}{\pi_1 + \pi_2}$ and $\omega_2 = 1 - \omega_1$ are normalized weights. Finally, the general form of fused pose $N(\theta, \Sigma)$ is computed (17) as

$$\begin{aligned} \Sigma^{-1} &= \omega_1 \Sigma_1^{-1} + \omega_2 \Sigma_2^{-1} \\ \theta &= \Sigma(\omega_1 \Sigma_1^{-1} \theta_1 + \omega_2 \Sigma_2^{-1} \theta_2) \end{aligned} \tag{15}$$

5 Experimental Results

In order to validate our approach, we recognized 3D objects in cluttered domestic environment. About 20 daily used domestic objects are employed for experiments, such as refrigerator, milk box, biscuit box, cup, book, etc. All images were captured with a stereo camera at a resolution of 640×480 pixels. 100 to 200 3D lines can be detected generally in an image. First of all, we test the proposed algorithm for both texture and textureless objects. Fig 5 illustrates recognition results to different kinds of selected objects. Poses are illustrated in both 3D space and their projection on 2D images. All of the objects are recognized correctly within seven top ranked interpretations.

In the interest of analyzing the reasonableness and effectiveness of probability computation, we choose two cluttered domestic scenes as shown in Fig. 6. In the first row of Fig. 6, several ambiguous interpretations are generated because the parallel line pair comes from two different objects (one line from cup and another line from box). In the second row of Fig. 6, multiple interpretations are generated from both target object and non-objects, because the non-objects

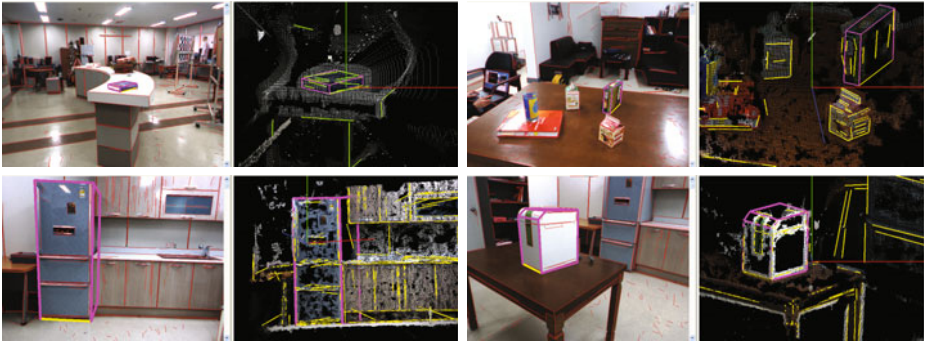


Fig. 5. Recognition results of four objects, results in both 2D image and 3D point clouds are illustrated. Textured objects (1st row): book and biscuit box, textureless objects(2nd row): refrigerator and kitchen refuse bin.

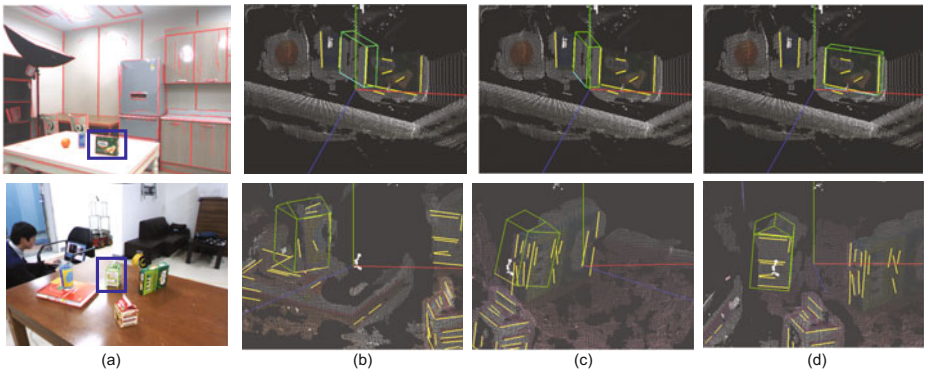


Fig. 6. Multiple interpretations generation and probability assignment. (a) is original 2D images, and two target objects are marked with rectangles. (b)-(d) are selected interpretations of each object, and only (d) is the correct recognition. First row demonstrates ambiguity from spurious features, the estimated probabilities of (b)-(d) are 0.392, 0.330 and 0.614 respectively. Second row shows ambiguity of other objects which have similar initial feature set as target object, the estimated probabilities of (b)-(d) are 0.469, 0.509 and 0.834 respectively, where the highest probability indicates the true object correctly. The figure is best viewed in color and PDF magnification.

have similar initial feature set with the target object. But our approach can discriminate these ambiguities by estimated probabilities. This demonstrates that initial interpretations are generated as a weak classifier without losing any possible candidates, and then can be verified probabilistically by additional evidences as a strong classifier to eliminate the spurious interpretations. Finally a small number of reasonable interpretations with high probability are selected.

In order to evaluate the performance of the proposed algorithm, three parameters are measured:

- **Detection probability β** : The probability that the complete set of generated interpretations contain at least one correct matching between the model and image.
- **Ranking index n** : Among all the interpretations ranked probabilistically, the ranking of the first correct interpretation leads to a true location of the target object.
- **Computation time t** : The average computation time for each model at each image.

Selected images shown in Fig. 7 are used to perform the experiment. For each specific object, there are three typical cases of scenarios from easy to difficult in domestic environment. In case 1, single object appears in the scene without any partial occlusion, which is the easiest case, as shown in first row of Fig. 7. In case 2, object with partial occlusion but no similar objects coexist, as shown in second row of Fig. 7. In case 3, also is the most difficult case, not only partial occlusion, but also similar objects are coexist with the target object, as shown in third row of Fig. 7. Actually, for each scenario of each object, 25 images are captured with different viewpoint, distance and illumination. The estimated value of parameter detection probability $\beta \equiv 1$ in all of three cases, thanks to the adopted weak initial feature that any possible candidates of target object will not lose. And the estimated values of ranking index n are shown in Fig. 8(a). Horizontal axis represents index of object model, where 20 objects are employed for this evaluation. From this we see that, for the simple scene which only include single object, the correct interpretation can be found only in top three interpretations. By examining just the top five interpretations for case 2 can get correct matching result. Even for the most difficult scene as case 3, where many similar objects are coexist with target object, the correct recognition can be achieved in just only top eight interpretations means that the probability computation is working as expected. Therefore, only a few interpretations with high probabilistic ranking

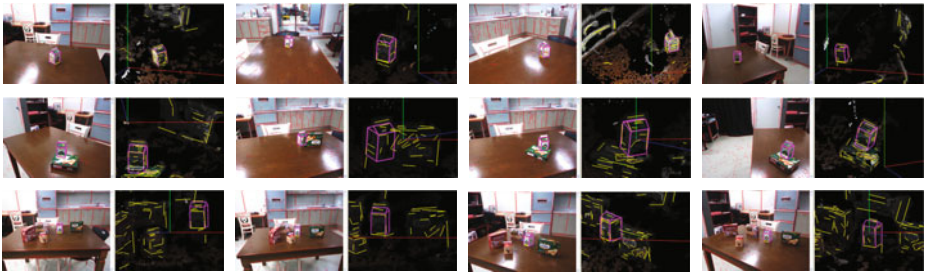


Fig. 7. Selected images for performance evaluation. 1st row shows case 1 with only single object in the foreground, but the background is still cluttered; 2nd row shows case 2, where the object is occluded with partial occlusion; 3rd row shows case 3, which is the most difficult case, not only partial occlusion, but also several similar objects are coexist with the target object. Both 2D and 3D recognition results are shown in images. The figure is best viewed in color and PDF magnification.

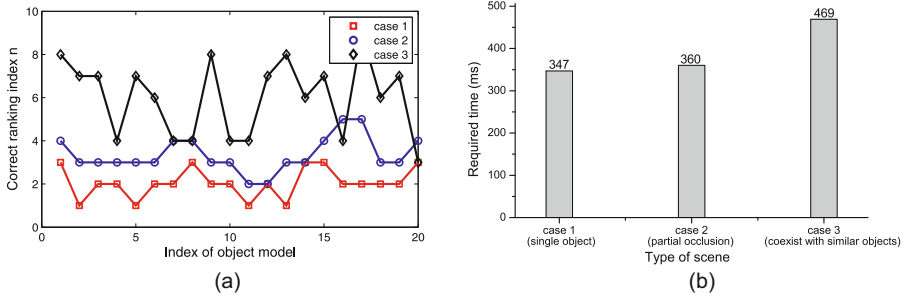


Fig. 8. Performance evaluation. (a) is the ranking index n , and (b) is computation time t .

need to be used in fusion stage. Finally, in this experiment, the proposed methods required averagely the computation time less than 500ms even for the very clutter scene, as is seen in Fig. 8(b).

6 Conclusion

3D object recognition and pose estimation are basic prerequisites for home service robotics. In this paper, a novel approach for probabilistic recognition based on multiple interpretations has been proposed, our approach represents the recognized object pose with probabilistic multiple interpretations, which are generated from invariant 3D features as parallel line pairs and perpendicular line pairs. An interpretation is represented as a weighted Gaussian *pdf* which is a region instead of a point in pose space. The probability of each interpretation is computed efficiently in terms of both likelihood and unlikelihood that is robust to occlusion and clutter. The top ranked interpretations are further verified and refined with a fusion strategy in a closed form. The fused interpretations are more confident with high probabilities, which can lead to more accurate pose estimation. Experiments show that the proposed approach can recognize object robustly in cluttered domestic environment in real time.

Acknowledgement. This work was supported by the MEST (Ministry of Education, Science and Technology), Korea, under the WCU (World Class University) program supervised by the KOSEF (Korea Science and Engineering Foundation) (R31-2008-000-10062-0), and by Priority Research Centers Program through the NRF (National Research Foundation of Korea) (2010-0020210). This work was also partially supported by the 21st Century Frontier Program (F0005000-2010-32), and in part by the KORUS-Tech Program (KT-2008-SW-AP-FSO-0004) funded by the MKE (Ministry of Knowledge Economy).

References

1. Roberts, L.G.: Machine perception of three-dimensional solids. In: Tipett, J.T. (ed.) *Optical and Electrooptical Information Processing*, pp. 159–197. MIT Press, Cambridge (1965)

2. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24, 381–395 (1981)
3. Beis, J.S., Lowe, D.G.: Indexing without invariants in 3d object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21, 1000–1015 (1999)
4. Costa, M.S., Shapiro, L.G.: 3d object recognition and pose with relational indexing. *Comput. Vis. Image Underst.* 79, 364–407 (2000)
5. David, P., Dementhon, D., Duraiswami, R., Samet, H.: Softposit: Simultaneous pose and correspondence determination. *International Journal of Computer Vision* 59, 259–284 (2004)
6. Vicente, M.A., Hoyer, P.O., Hyvarinen, A.: Equivalence of some common linear feature extraction techniques for appearance-based object recognition tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29, 896–900 (2007)
7. Bicego, M., Castellani, U., Murino, V.: A hidden markov model approach for appearance-based 3d object recognition. *Pattern Recognition Letters* 26, 2588–2599 (2005)
8. Cyr, C.M., Kimia, B.B.: A similarity-based aspect-graph approach to 3d object recognition. *International Journal of Computer Vision* 57, 5–22 (2004)
9. Min, S., Hao, S., Savarese, S., Li, F.F.: A multi-view probabilistic model for 3d object classes. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009*, pp. 1247–1254 (2009)
10. Johnson, A.E., Hebert, M.: Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21, 433–449 (1999)
11. Frome, A., Huber, D., Kolluri, R., Bulow, T., Malik, J.: Recognizing objects in range data using regional point descriptors. In: Pajdla, T., Matas, J.(G.) (eds.) *ECCV 2004*. LNCS, vol. 3023, pp. 224–237. Springer, Heidelberg (2004)
12. Ohbuchi, R., Osada, K., Furuya, T., Banno, T.: Salient local visual features for shape-based 3d model retrieval. In: *IEEE International Conference on Shape Modeling and Applications, SMI 2008*, pp. 93–102 (2008)
13. Shimshoni, I., Ponce, J.: Probabilistic 3d object recognition. *International Journal of Computer Vision* 36, 51–70 (2000)
14. David, P., DeMenthon, D.: Object recognition in high clutter images using line features. In: *Tenth IEEE International Conference on Computer Vision, ICCV 2005*, vol. 2, pp. 1581–1588 (2005)
15. Zhaojin, L., Seungmin, B., Sukhan, L.: Robust 3d line extraction from stereo point clouds. In: *IEEE Conference on Robotics, Automation and Mechatronics*, pp. 1–5 (2008)
16. Bregler, C., Malik, J.: Tracking people with twists and exponential maps. In: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1998*, pp. 8–15 (1998)
17. Genest, C., Zidek, J.V.: Combining probability distributions: A critique and an annotated bibliography. *Statistical Science* 1, 114–135 (1986)

Planar Affine Rectification from Change of Scale

Ondřej Chum and Jiří Matas

CMP, Dept. of Cybernetics, Faculty of EE, CTU in Prague

Abstract. A method for affine rectification of a plane exploiting knowledge of relative scale changes is presented. The rectifying transformation is fully specified by the relative scale change at three non-collinear points or by two pairs of points where the relative scale change is known; the relative scale change between the pairs is not required. The method also allows homography estimation between two views of a planar scene from three point-with-scale correspondences.

The proposed method is simple to implement and without parameters; linear and thus supporting (algebraic) least squares solutions; and general, without restrictions on either the shape of the corresponding features or their mutual position.

The wide applicability of the method is demonstrated on text rectification, detection of repetitive patterns, texture normalization and estimation of homography from three point-with-scale correspondences.

1 Introduction

The problem of affine rectification of a plane, *i.e.* the problem of transforming an image by a homography so that the vanishing line of the plane becomes the line at infinity, arises in many applications, *e.g.* in document processing [1,2], detection of repetitive structures [3] and texture analysis [4,5]. The plane of interest appears in the rectified images as if viewed by an affine camera, *i.e.* projected by a set of parallel rays and scaled. The restoration of affine properties like parallelism and global scale simplifies subsequent application-dependent processing steps like geometric normalization, detection and recognition.

In the paper, a general yet simple method for affine rectification of a plane is introduced. The algorithm exploits knowledge of relative scale changes in the local neighbourhood of image points lying in the plane. The rectifying transformation is fully specified by the relative scale change at three non-collinear points. Another minimal case covered by the method applies in the situation where for two pairs of points the relative scale change is known; the relative scale change between the pairs is not required.

A situation in which the relative scale change is known at different points arises often in practice. Consider, *e.g.* the problem of affine rectification of a repeated pattern on a planar surface, say a facade, Fig. 1. In a perspective image of the facade, the features detected on the windows in general vary in size (area). In reality, it is common that (at least some of) the windows are of the same size. The task addressed in the paper is to find a planar homography H that transforms the image of the facade so that all the window features cover the

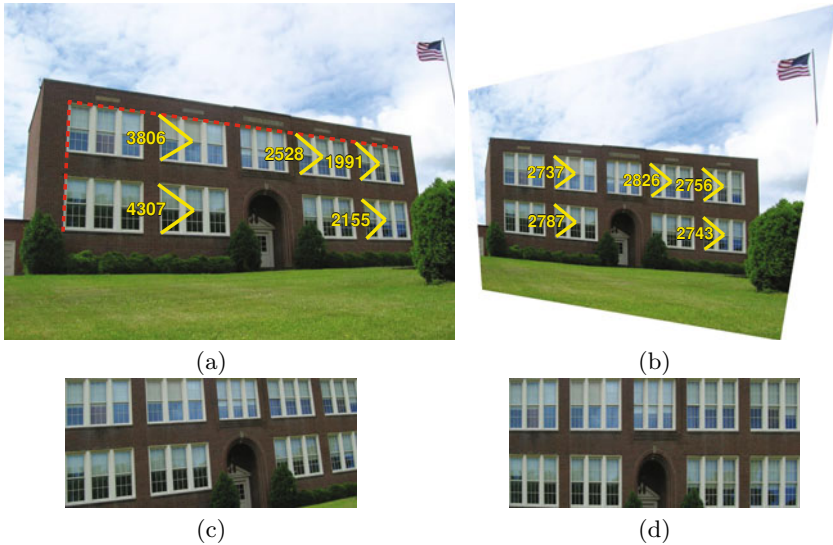


Fig. 1. Affine rectification. Original image (a) - the area of the triangular patches differs from 1991 to 4307 pixels due to the perspective projection. Rectified image (b) - the areas are approximately the same, as in reality. Parallel lines on the facade are not parallel in the original image (c), and are parallel after the normalization (d). The cut-outs (c) and (d) are parallelograms defined by two red line segments in (a).

same area. We show in the experimental section that the method is applicable in many situations.

The proposed method has the following advantages: (generality) no assumptions are made about either the shape of the features or their mutual position; features need not lie on a regular grid nor on lines and may be arbitrarily rotated; (stability) the rectification is computed from ratios of areas, a very stable property insensitive to many image degradations such as discretization; (simplicity) the rectification algorithm is simple, easy to implement and without parameters; (linearity) the constraints on the scale change are expressed as linear constraints on the entries of the homography matrix H that represents the transformation. Linear constraints are very convenient as they can be used with minimal sets (in RANSAC-like [6] robust estimators) as well as in (algebraic) least squares solutions from all available data.

The derivation of the algorithm assumes that the features are sufficiently small so that their scale change reasonably approximates the scale change (of an infinitesimal patch) at corresponding points. Such an assumption is made by wide-baseline matching approaches using affine covariant feature points and/or affine invariant feature descriptors. We show experimentally in Sec. 3 that the assumption holds in practice.

Previous work. Affine rectification algorithms proposed in the literature differ by the assumptions about the structures present in the image that are exploited in the process. The most straightforward approaches detect two distinct vanishing points [7].

The problem of vanishing line detection has been addressed for elements repeated by translation on a plane. The geometric relation of the elements after projective transformation is called *elation* [7]. A comprehensive study of vanishing line (and points) detection based on the elation assumption is given in [3]. Another approach exploiting elations for detection of vanishing line in a projective image of a texture was proposed in [8]. Other approaches, specially in the text analysis, assume, that parallel lines with equal spacing can be detected in the image. The normalization (vanishing line) is then estimated from the intersection of the parallel lines and a cross-ratio of collinear set of points on those lines [1].

Publications on affine rectification have appeared in the field of shape-from-texture [9]. In general, assuming homogeneity of the texture, more complex structure than orientation of a plane can be estimated [10]. However, a fairly complex optimization approach is necessary in this case. There are many approaches to vanishing point and/or line detection from the texture. Voting schemes based on dominant direction of the texture can be used to determine a vanishing point [11]. In [12], another voting scheme based on distortion of the power spectrum under projective transformation is used to detect the vanishing line.

Similar idea to ours has appeared in Ohta's 1981 paper [13] on shape from texture. Despite the different derivations the results are closely related. In fact the formulation in [13] is a special case of ours. Our formulation allows to extend the applicability of the idea beyond a planar rectification, for example to multi-view geometry. Our derivation yields a single linear constraint per feature while Ohta's approach produces one linear constraint per a pair of textured regions. Finally, we show significantly higher applicability than [13] or its extension [14], for example the features of interest (or texture) does not have to cover the whole image.

The rest of the paper is organized as follows. First, the method is derived in sections 2 and 2.1. Extension to multiple independent feature sets is introduced in section 2.2. Experiments and applications of the proposed method to various tasks are presented in section 3: simple examples of the minimal cases 3.1, text rectification 3.2, non-linear repeated structures 3.3, segmentation of multiple planes with repeated pattern 3.4, texture rectification 3.5, and experiments on synthetic data 3.6 and 3.7. The applicability of the approach to image to image homography estimation from point-with-scale correspondences is discussed in section 3.8. Conclusions are drawn in section 4. A proof of degenerate case of collinear points can be found in an appendix.

2 The Method

First, the concept of local scale change under planar homography is introduced and its properties are discussed. Next, a decomposition of a homography simplifying the algebra is presented. Finally, it is shown that constraints on the local scale change under planar homography (*i.e.* perspective transformation of a plane) lead to linear constraints on the entries of the homography matrix.

Homography is a mapping from a projective plane P^2 to P^2 and it is commonly represented by a (homogeneous) matrix H , or equivalently, by

inhomogeneous pair of functions (h_x, h_y) [7]. In this section, we restrict the homographies to be in the following form

$$\mathbf{H} = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & 1 \end{pmatrix} \quad \text{or} \quad \begin{aligned} h_x(x, y) &= \frac{h_1x + h_2y + h_3}{h_7x + h_8y + 1}, \\ h_y(x, y) &= \frac{h_4x + h_5y + h_6}{h_7x + h_8y + 1}. \end{aligned} \quad (1)$$

The sufficiency of the $\mathbf{H}_{3,3} = 1$ parametrization is discussed and justified in section 2.1. The first order Taylor expansion at point (x, y) and the Jacobian $\mathbf{J}_\mathbf{H}$ locally approximating the homography

$$h(x + \delta_x, y + \delta_y) \approx \begin{pmatrix} h_x(x, y) \\ h_y(x, y) \end{pmatrix} + \mathbf{J}_\mathbf{H}(x, y)d_{xy} \quad (2)$$

is an affine transformation for which the concept of scale change is well defined. The local scale change at point (x, y) under the perspective transformation is thus defined as the scale change of the first order, *i.e.* affine, approximation at point (x, y)

$$s(\mathbf{H}, x, y) = \det(\mathbf{J}_\mathbf{H}(x, y)). \quad (3)$$

Any homography \mathbf{H} in the form of (1) can be decomposed into a product $\mathbf{A}\hat{\mathbf{H}}$ of an affine transformation \mathbf{A} and a homography $\hat{\mathbf{H}}$ as follows

$$\begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & 1 \end{pmatrix} = \begin{pmatrix} h_1 - h_3h_7 & h_2 - h_3h_8 & h_3 \\ h_4 - h_6h_7 & h_5 - h_6h_8 & h_6 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ h_7 & h_8 & 1 \end{pmatrix} \quad (4)$$

It can be shown that the scale change of homography \mathbf{H} expressed in terms of \mathbf{A} and $\hat{\mathbf{H}}$ is

$$s(\mathbf{A}\hat{\mathbf{H}}, x, y) = \det(\mathbf{A})s(\hat{\mathbf{H}}, x, y).$$

The advantage of the decomposition (4) is that the influence of parameters $h_1 \dots h_6$ on the local scale change is reduced to a single global (*i.e.* position-independent) parameter $\det \mathbf{A}$ in the expression

$$s(\mathbf{H}, x, y) = \det(\mathbf{A})s(\hat{\mathbf{H}}, x, y) = \det(\mathbf{A}) \det(\mathbf{J}_{\hat{\mathbf{H}}}(x, y)). \quad (5)$$

The determinant of the Jacobian of the matrix $\hat{\mathbf{H}}$ at (x, y) is

$$\det(\mathbf{J}_{\hat{\mathbf{H}}}(x, y)) = \det \left((h_7x + h_8y + 1)^{-2} \begin{pmatrix} h_8y + 1 & -xh_8 \\ -yh_7 & h_7x + 1 \end{pmatrix} \right) = (h_7x + h_8y + 1)^{-3}.$$

Setting $\det(\mathbf{A}) = \alpha^3$ and substituting into equation (5), we get

$$s(\mathbf{H}, x, y) = \alpha^3(h_7x + h_8y + 1)^{-3}. \quad (6)$$

After re-arranging the equation, a constraint linear in h_7 , h_8 , and α is obtained:

$$(x \ y \ -s(\mathbf{H}, x, y)^{-1/3}) \begin{pmatrix} h_7 & h_8 & \alpha \end{pmatrix}^\top = -1. \quad (7)$$

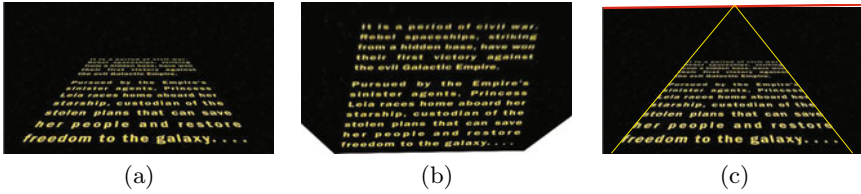


Fig. 2. Star Wars credits: (a) original image, (b) estimated (up to an affine transformation) normalized image, (c) original image with the estimated vanishing line (red) and manually drawn the parallel margin lines (yellow)

Three point locations (x_i, y_i) and the corresponding local scale changes $s(H, x_i, y_i)$ are required to estimate the homography \hat{H} . Any composition of affine transformation A , $\det(A) = \alpha^3$ and the homography \hat{H} , *i.e.* $H = A\hat{H}$ will satisfy the constraints on the local scale change. The vanishing line l in the source image is the pre-image of the line at infinity $(0\ 0\ 1)^T$

$$l = H^T(0\ 0\ 1)^T = \hat{H}^T A^T(0\ 0\ 1)^T = \hat{H}^T(0\ 0\ 1)^T = (h_7\ h_8\ 1)^T. \tag{8}$$

If $p, p > 3$, points with the local scale change are available, the least squares method is applicable. The data matrix $Z \in R^{p \times 3}$ is composed of rows

$$Z = \begin{pmatrix} x_i & y_i & -s(H, x_i, y_i)^{-1/3} \\ \vdots & \vdots & \vdots \end{pmatrix}, \tag{9}$$

one per each point (x_i, y_i) . The solution is then obtained as

$$(h_7\ h_8\ \alpha)^T = -Z^\dagger \mathbf{1}^{1 \times p}, \tag{10}$$

where Z^\dagger is pseudo-inverse of Z and $\mathbf{1}^{1 \times p}$ is a column vector of p ones.

In many applications, the scale change is not interesting or not known and only relative scale changes at different points are known. Here, the estimated parameter α can be simply ignored. This is *e.g.* the case for the facade example Fig. 1, where the windows are assumed to have the same, but unknown, real size. In such cases, the $s(H, x, y)$ is multiplied by an unknown scalar.

2.1 The Choice of Parametrization

The chosen parametrization of matrix H in section 2 does not cover all possible homographies. Namely, it does not include the set of homographies \mathcal{H}_0 with $H(3, 3) = 0$, *i.e.* homographies that map the origin of the image coordinate system $(0\ 0\ 1)^T$ to a point at infinity. Hence, if it is possible to choose the origin so that it is guaranteed that the required solution does not map the origin to infinity, the $H(3, 3) = 1$ parametrization is correct.

A frequent choice of the origin of the image coordinate system – the (top left) corner of the image – does not always guarantee the above described property. In particular, in Fig. 2, the top left corner lies on the vanishing line mapped to a line at infinity by the affine rectifying homography.



Fig. 3. An example of multiple features on an element of repeated pattern

The origin must not lie on the vanishing line, as the estimated transformation sends the vanishing line to infinity. Since the algorithm is used for affine rectification which is equivalent to detection of the vanishing line, based on scale change of measured features, good candidates for the origin are the measured points. This stems from the fact, that the point and its relative finite scale change could not have been measured at the line at infinity.

More generally, since the traditional (directional) camera sees only points in front of the camera [15], the vanishing line cannot ‘cut through’ the observed points. Therefore, any point inside the convex hull of the observed points will serve well as the origin of the coordinate system. The centre of gravity of the observed points was used in our implementation.

Note on the data normalization. In the least squares problem, some algebraic error (with no direct geometric meaning) is minimized. It has been shown that in such problems, it is advantageous to normalize the data points so that the elements of the measurement matrix Z have similar magnitudes [16]. Choosing the origin at the centroid of the data, re-scaling the data and suitable selection of the relative scale change prior to evaluation eqn. (10) can be used to stabilize the least squares solution.

2.2 Extension to Multiple Independent Sets

As mentioned above, often only the relative scale change between a set of points is known. This section addresses the situation where multiple such sets are available. The relative scale change is known within each set, the relations between different sets is unknown.

As an example, let us have a look at the repetitive structures again. In general, the features detected in the image do not correspond one to one to the repeated elements. Typically, each element is covered by a number of features, as in Fig. 3. This number is also varying, as the repeatability of the features (as well as the stability of the descriptors) is not perfect. For each individual set of matching features, one can hypothesize that these are of the same size in reality, since else it is unlikely for the appearance of two patches to match. However, the area ratio of different features is not known in general.

For the sake of clarity, the derivation is demonstrated for two sets only. The extension to a general number of such sets is straightforward. There are two unknowns h_7 and h_8 shared between all the sets. Each set introduces an additional variable α_k . The variable represents the relative scale change of the whole set with respect to other sets. The equations are then arranged in the same way as in equation (7):

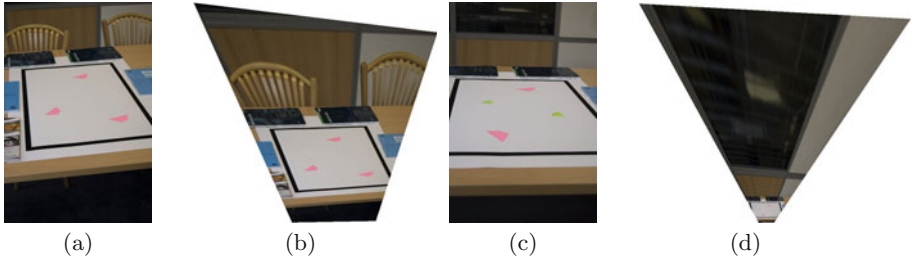


Fig. 4. Toy examples for the minimal cases of three points (a-b) and two plus two points (c-d)

$$\begin{pmatrix} x_i & y_i & -s(\mathbf{H}, x_i, y_i)^{-1/3} & 0 \\ x_j & y_j & 0 & -s(\mathbf{H}, x_j, y_j)^{-1/3} \end{pmatrix} (h_7 \ h_8 \ \alpha_1 \ \alpha_2)^\top = \begin{pmatrix} -1 \\ -1 \end{pmatrix}. \quad (11)$$

Each feature in a set adds one constraint, at least two features have to be available for each set to add more constraints than unknowns. In general, if there are p points in q sets, there are $2 + q$ unknowns and p constraints. For two sets, two points per set are sufficient to estimate the rectifying transformation. For an example, see section [3.1](#)

3 Experiments

In this section a variety of experiments with different settings are presented.

3.1 Toy Example

Two images of coplanar patches – Fig. [4\(a\)](#) and (c) – are used to demonstrate the minimal cases described in sections [2](#) and [2.2](#). Very simple colour segmentation was used to locate the pink and green patches. The patches were represented by their location (the centre of gravity) and the scale (the number of pixels occupied by the patches). To simulate the two cases of minimal sets, the experiments was designed as follows: 1. the pink patches are of the same size, 2. the green patches are of the same size, and 3. the relative sizes of the pink and green patches are unknown.

The rectified images – Fig. [4\(b\)](#) and (d) – show that the part of the scene that has been reduced by the projective transformation (further away from the camera) is expanded by the normalization. Also note that after the normalization, the parallel lines on the sheet of paper are again (very close to) parallel.

3.2 Text Rectification

In text localization and recognition in photographs taken in unconstrained conditions, geometric rectification is performed before classification of characters. The algorithm proposed in the paper is significantly simpler and more general

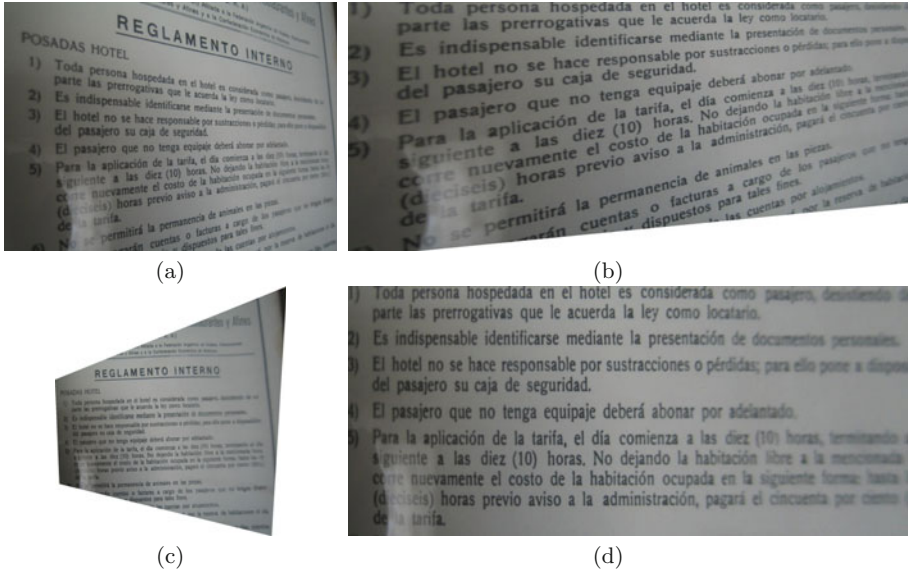


Fig. 5. Text rectification. (a) original image, (b) manual rectification using an affine transformation, (c) automatic affine rectification (d) manual rectification using an affine transformation after removing the perspective.

than the approaches commonly used in document processing for affine rectification, *e.g.* [2] who requires a reliable procedure for fitting a baseline and topline of the text.

Applicability of the proposed procedure to the text rectification problem is demonstrated in Fig. 5. The top-left image (b) shows that affine normalization, is insufficient, non-parallel lines in the original image (a) say non-parallel. Fig. 5(c) shows the results of the proposed algorithm. The correspondences necessary for estimation of the relative scale are obtained fully automatically on identical characters by the MSER+LAF method [17]. Outliers and out-of-plane pairs are removed by RANSAC. The rectification based on tens of scale ratios is quite precise, see Fig. 5(d) which is an affine transformation of the rectified of image (c). The final affine rectification was done manually as it is not the topic of the paper - the proposed algorithm has no concept of a line of text or left margin; an example of an automatic method is in [2].

3.3 Darts

The "Darts" image, Fig. 6, is an example where direct detection of vanishing points and hence the vanishing line is difficult. The dominant linear features on the board that intersect in the bull's eye have different orientations and intersect the vanishing line in different ideal points.

The proposed method estimates the rectifying transformation from multiple sets of corresponding features, Fig. 6(e). Notice that the correspondences are between features with different orientations and not lying on straight lines. After

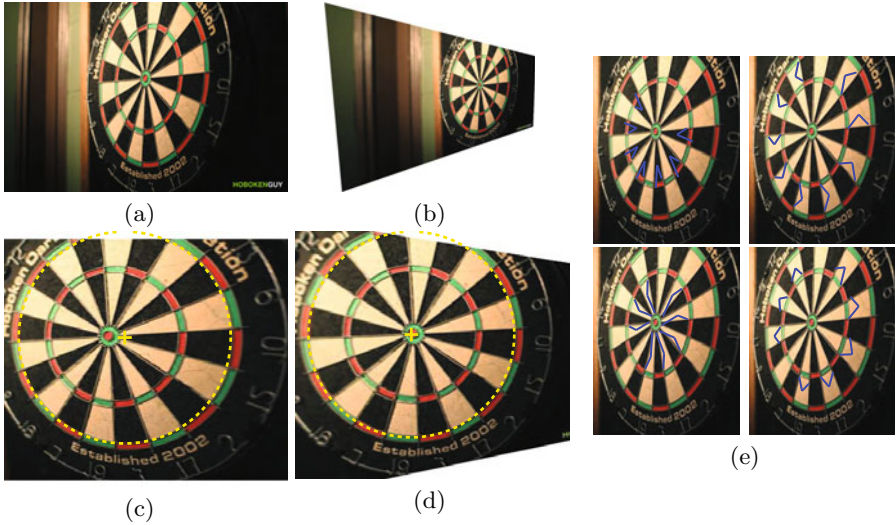


Fig. 6. Darts: (a) the original image; (b) automatic rectification; (c) manual rectification to a circle (dashed, centre labelled with '+') by an affine transformation from the original image, (d) the rectified image; (e) some of the matching feature groups superimposed

(manually) mapping the the ellipse corresponding to the inner rim of the double scoring area, a fronto-parallel view of the board is obtained. The centre of the dotted yellow circle is very close to the centre of the bulls eye.

A direct affine mapping of ellipse corresponding to the inner rim in (a) to a circle results in image (c), but this view does not correspond to a fronto-parallel view of the board.

3.4 Segmentation of Multiple Planes with Repeated Pattern

The proposed method is not restricted to a single planar rectification. With RANSAC, a robust estimator, it is possible to separate features on a single plane from outliers. In the presence of multiple models (in our case multiple planes), consecutive execution of RANSAC with removal of features consistent with detected model [18] provides an efficient strategy.

In the simplest case, the two (or multiple) planes would not share features (different buildings, etc.). The example in Fig. 7 is more challenging, it shows that even if the planes share a common repetitive pattern and therefore the MSER+LAF method establishes correspondences between the two planes, the geometric constraints on the relative scale change are sufficiently discriminative to segment the planes.

3.5 Textures

The proposed method is also applicable to irregular statistical textures. In Fig. 8, an example of affine texture rectification is shown. For statistical textures, the



Fig. 7. Two planar surfaces with a repetitive pattern segmented by RANSAC. Left: one group of matching features, inliers to one model in red, the other in green, outliers to both models in yellow. Right: the convex hull of consistent features.



Fig. 8. Texture rectification: the original image (left), affine rectification (middle), and the rectified texture (right)

MSER and LAF method is not suitable since it requires that corresponding regions are geometrically close to identical. In the example, an affine covariant elliptical region detector [19] together with the SIFT descriptor [20] was used.

3.6 Scale Change from Local Patches

One of the inputs of the proposed method is a scale change of an infinitesimal patch. However, it is typically only possible to measure the scale (change) at image patches that are of area of tens of pixels. In the first experiment, we measure how the estimate of the scale change affects the results.

First, A pattern of 5×5 local affine patches is generated. It is transformed by a homography with varying values of h_7 and h_8 . The pattern is then resized and translated to fit a 800×600 image. Examples of four patterns are shown in Fig. 9. All situations in the experiment from the ‘convex hull’ of these four examples.

Each synthetic image was processed as follows. Each local affine patch was represented by the centre of gravity of the triangle and by the scale (area) of the triangle. A normalizing homography that transforms all patches to equal scale was estimated using the proposed method. In an ideal case, when the infinitesimal scale change is estimated exactly, all transformed patches would have exactly the same scale. The ratio of maximal resulting scale to the minimal resulting scale was recorded for each parameter setting. The results are visualized in Fig. 10.

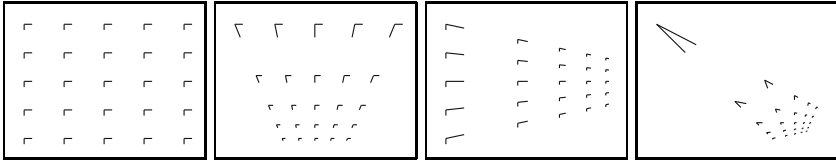


Fig. 9. Four examples of different levels of perspective deformations used in the synthetic experiments. All images are 800×600 pixels.

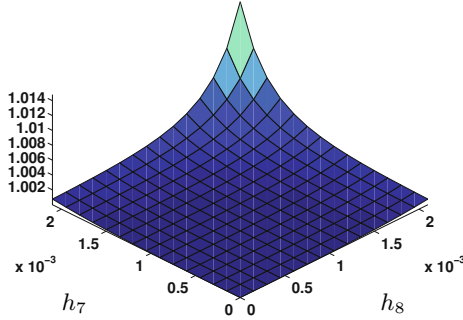


Fig. 10. Scale error after affine rectification. Ratio of the largest and the smallest feature after rectification to equal size.

It can be seen that even for extreme perspective deformations, the local scale is estimated sufficiently precisely and the ratio of areas of the largest and the smallest normalized patches is close to one. If necessary, the procedure can be iterated to eliminate the effect of the inaccurate estimation of the scale change. In the above experiment, after first iteration, the scale ratio of the areas of the largest to the smallest normalized patches was one up to numerical precision.

3.7 Sensitivity to Noise

This experiment also uses the settings from Fig. 9. Here, the transformed patches (the coordinates of the triangle corner points) were corrupted by additive Gaussian noise with $\sigma = 1.5$ pixels. Robust rectifying homography estimation via RANSAC was applied and three quantities were measured. First, how well the estimated homography rectifies the noise-less patches. The number of correctly rectified noise-less patches (the scale change error below 1.1) is shown in Fig. 11(a). Second, the number of RANSAC inliers is shown in Fig. 11(b). The number of inliers is well correlated with the number of correctly rectified noiseless patches. Third, the average scale error on all noiseless patches (not only inliers) is depicted in Fig. 11(c). All plots are averages over 50 executions of RANSAC.

3.8 Image to Image Homography

Another straightforward application of the proposed method is the estimation of image to image planar homography from scale-covariant features, such as the

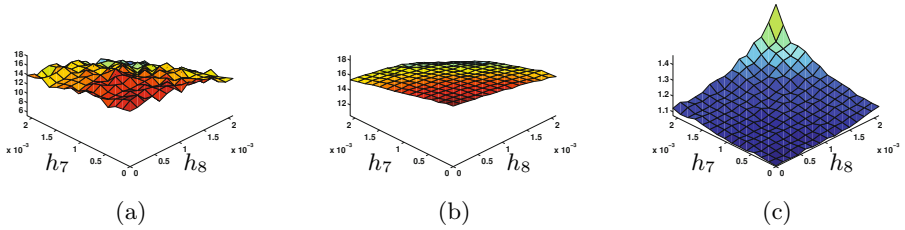


Fig. 11. Affine rectification estimated from regions corrupted by noise. Each plot shows 225 different settings of the parameters projective deformation h_7 and h_8 : (a) noiseless features with scale error below 1.1, (b) number of RANSAC inliers, (c) average scale error on noiseless features; results averaged over 50 executions.



Fig. 12. Graffiti images

Table 1. Comparison of the image to image homography estimation from samples of four point correspondences and three point-with-scale correspondences. The number of tentative correspondences (‘tentative’), percentage of inliers detected by the methods (‘% 4-inliers’ and ‘% 3-inliers’), the number of samples required in RANSAC (‘4-samples’ and ‘3-samples’), and the number of scale consistent samples in the point-with-scale method (‘3-valid’). Results averaged over 50 executions.

image pair	tentative	% 4-inliers	% 3-inliers	4-samples	3-samples	3-valid
1-2	877	61.09	61.13	31.4	18.3	6.3
1-3	694	33.81	32.55	356.7	151.3	19.8
1-4	493	12.48	10.99	20861.0	4207.8	148.9
2-3	988	52.98	52.42	57.0	30.5	8.8
2-4	732	30.17	28.57	565.0	209.9	22.2
3-4	1043	61.74	61.27	30.1	18.2	5.2

DoG [20]. Only three correspondences are required to estimate the full projective homography. First, the projective part \hat{H} is estimated from the scale change between the tree corresponding features. The affine part A is then given by the coordinates of the corresponding features in the two images. Sampling three instead of four points in RANSAC speeds up the robust estimation process, if scale information is available which is the case for scale and affine covariant features.

Furthermore, one non-linear constraint is available. It is not used in the estimation and can be used to verify that a homography matching the three point-with-scale correspondences induces the correct scale change. This constraint is

the scale of the affine transformation \mathbf{A} . The scale of the affine part, given by $\det(\mathbf{A})$, is obtained during the estimation of the projective part as α^3 in eqn. (7). Using the constraint in RANSAC, a number of contaminated samples can be rejected without the necessity of calculating consensus set size.

Images used in the experiment are a subset of a standard dataset [19], see Fig. 12. A combination of DoG features with the SIFT descriptor [20] was used, followed by a RANSAC with a local optimization step [21]. The comparison of the number of RANSAC samples in the homography estimation is shown in Table 1; results were averaged over 50 executions. The results show that using three point-with-scale correspondences allows to estimate the homography in significantly lower number of samples. If the scale consistency check is applied (the threshold was set to 1.1 in the experiments), the consensus is computed for only a small fraction of the samples – the last column of Table 1. On the other hand, the three point-with-scale samples provide a little less stable performance (slightly lower average of estimated inlier ratios) than four point correspondences.

4 Conclusions

A simple yet generally applicable method for affine rectification of a plane exploiting knowledge of relative scale changes was presented. The method also allows estimating the homography between two views of a planar scene from three point-with-scale correspondences. A significant speed-up was achieved w.r.t. the standard four point procedure.

The utility of the method was demonstrated on text rectification, detection of repetitive patterns, texture normalization and estimation of homography from three points-with-scale correspondences.

Acknowledgement. The authors were supported by GAČR project 102/09/P423, by EC project ICT-215078 DIPLECS and by ČVUT SGS10/069/OHK3/1T/13.

References

1. Clark, P., Mirmehdi, M.: Rectifying perspective views of text in 3d scenes using vanishing points. *Pattern Recognition* 36, 2673–2686 (2003)
2. Myers, G.K., Bolles, R.C., Luong, Q.T., Herson, J.A., Aradhya, H.: Rectification and recognition of text in 3-d scenes. *IJDAR* 7, 147–158 (2005)
3. Schaffalitzky, F., Zisserman, A.: Planar grouping for automatic detection of vanishing lines and points. *Image and Vision Computing* 18, 647–658 (2000)
4. Ribeiro, E., Hancock, E.R.: 3-d planar orientation from texture: Estimating vanishing point from local spectral analysis. In: Carter, J.N., Nixon, M.S. (eds.) *BMVC*, British Machine Vision Association (1998)
5. Lelandais, S., Boutté, L., Plantier, J.: Shape from texture: Local scales and vanishing line computation to improve results for macrotexels. *Int. J. Image Graphics* 5, 329–350 (2005)
6. Fischler, M., Bolles, R.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *CACM* 24, 381–395 (1981)

7. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004) ISBN: 0521540518
8. Criminisi, A., Zisserman, A.: Shape from texture: homogeneity revisited. In: Proc. BMVC, UK, pp. 82–91 (2000)
9. Witkin, A.: Recovering surface shape and orientation from texture. Artificial Intelligence 17, 17–45 (1981)
10. Forsyth, D., Ponce, J.: Computer Vision: A Modern Approach. Prentice-Hall, Englewood Cliffs (2003)
11. Rasmussen, C.: Texture-based vanishing point voting for road shape estimation. In: Proc. BMVC (2004)
12. Ribeiro, E., Hancock, E.: Estimating the perspective pose of texture planes using spectral analysis on the unit sphere. Pattern Recognition 35, 2141–2163 (2002)
13. Ohta, Y., Maenobu, K., Sakai, T.: Obtaining surface orientation from texels under perspective projection. In: IJCAL, Vancouver, Canada, pp. 746–751 (1981)
14. Aloimonos, Y.: Shape from texture. Biological Cybernetics 58, 345–360 (1988)
15. Hartley, R.: Chirality. IJCV 26, 41–61 (1998)
16. Hartley, R.: In defence of the 8-point algorithm. In: ICC, vol. 95, pp. 1064–1070 (1995)
17. Obdržálek, Š., Matas, J.: Object recognition using local affine frames on distinguished regions. In: Proc. BMVC, pp. 113–122 (2002)
18. Torr, P.H.S.: Outlier Detection and Motion Segmentation. PhD thesis, Dept. of Engineering Science, University of Oxford (1995)
19. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L.: A comparison of affine region detectors. IJCV 65, 43–72 (2005)
20. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60, 91–110 (2004)
21. Chum, O., Matas, J., Obdržálek, Š.: Enhancing RANSAC by generalized model optimization. In: Proc. of the ACCV, vol. 2, pp. 812–817 (2004)

Appendix: Degenerate Case

Assume three collinear points (x, y) , $(x + \alpha d_x, x + \alpha d_y)$, and $(x + \beta d_x, y + \beta d_y)$. Let the h_7 and h_8 be the parameters of the decomposition of the normalizing homography by eqn. (1). Then, the 3×3 data matrix \mathbf{Z} from eqn. (9) has the following form

$$\mathbf{Z} = \begin{pmatrix} x & y & h_7x + h_8y + 1 \\ x + \alpha d_x & y + \alpha d_y & h_7(x + \alpha d_x) + h_8(y + \alpha d_y) + 1 \\ x + \beta d_x & y + \beta d_y & h_7(x + \beta d_x) + h_8(y + \beta d_y) + 1 \end{pmatrix}$$

The matrix \mathbf{Z} is singular with vector \mathbf{n}

$$\mathbf{n} = (-h_7x d_y + h_7d_x y - d_y, h_8y d_x - x h_8 d_y + d_x, x d_y - d_x y)^\top$$

spanning the null space of \mathbf{Z} . The vector $\mathbf{h} = (h_7, h_8, 1)^\top$ solves the equation $\mathbf{Z}\mathbf{h} = -\mathbf{1}$. Hence, there is a one-dimensional family of solutions $\mathbf{h} + \lambda \mathbf{n}$. It corresponds to a pencil of lines $\mathbf{h} + \lambda \mathbf{n}_0$, where

$$\mathbf{n}_0 = (-h_7x d_y + h_7d_x y - d_y, h_8y d_x - x h_8 d_y + d_x, 0)^\top.$$

All lines in the pencil pass through a point $\mathbf{h} \times \mathbf{n}_0$, which is the vanishing point lying on a line given by the collinear points.

Sensor Measurements and Image Registration Fusion to Retrieve Variations of Satellite Attitude

Régis Perrier¹, Elise Arnaud², Peter Sturm¹, and Mathias Ortner³

¹ INRIA Grenoble

² Université Joseph Fourier, LJK

³ EADS Astrium

Abstract. Observation satellites use pushbroom sensors to capture images of the earth. These linear cameras acquire 1-D images over time and use the straight motion of the satellite to sweep out a region of space and build 2-D images. The stability of the imaging platform is crucial during the acquisition process to guaranty distortion free images. Positioning sensors are used to control and rectify the attitude variations of the satellite, but their sampling rate is too low to provide an accurate estimate of the motion. In this paper, we describe a way to fuse star tracker measurements with image registration in order to retrieve the attitude variations of the satellite. We introduce first a simplified motion model where the pushbroom camera is rotating during the acquisition of an image. Then we present the fusion model which combines low and high frequency informations of respectively the star tracker and the images; this is embedded in a Bayesian setting. Lastly, we illustrate the performance of our algorithm on three satellite datasets.

1 Introduction

Currently, most of the remote sensing applications for observing the earth use pushbroom cameras. Such sensors became popular in the late 1970s with linear CCD as a way to get high resolution images [1]. Nowadays, they are still highly in use for their robustness against space turbulence, at a lower cost and higher resolution than classical 2-D CCD sensors.

In its principle, this linear sensor is mounted on a moving platform and captures 1-D image over time. When the platform is moving straight perpendicular to the axis of the camera, the pushbroom sensor sweeps out a region of space and build a complete 2-D image, this acquisition process is resumed in figure 1. This camera has already been described several times [2,3,4].

Such linear arrays are manufactured in very large quantities for use as imagers in document scanners, fax machines, bar code readers or hand-held scanners. In those cases, the stability of the platform is either not critical, or controlled by motors and stabilizers. In the space context, the satellite is moving on its orbit and may be subject to space turbulence and other physical effects which make

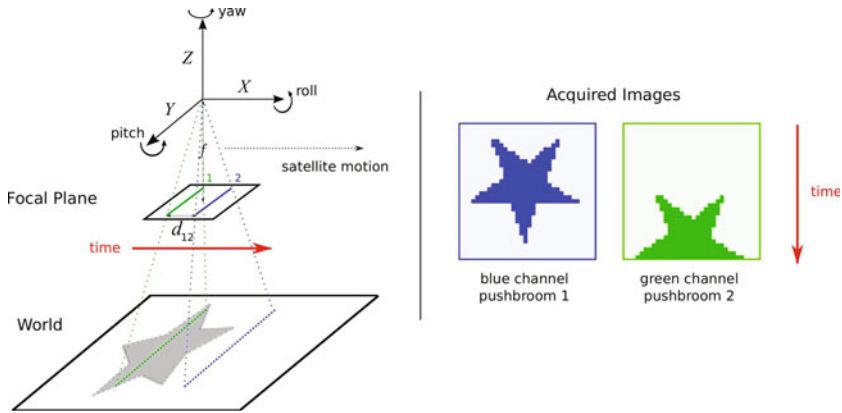


Fig. 1. Pushbroom acquisition principle: cameras 1 and 2 are moving straight along the X axis and recording 1-D images over time; Y is the camera axis and Z the orthogonal axis to the focal plane. We define the orientation of the camera with the yaw (rotation around Z), the roll (rotation around X) and the pitch (rotation around Y).

it deviating from its trajectory. Constancy on the attitude¹ is crucial during the acquisition of an image strip. Small rotations of the pushbroom camera over time warp each 1-D image and consequently the whole 2-D image; figure 2a presents a synthetic example of such warps. Attitude variations of the satellite need to be recorded to ensure the satellite control and to rectify the images if needed.

In order to build color images, several pushbroom cameras of different modalities are set in parallel onto the focal plane; figure 2b shows a typical focal plane of an observation satellite. As the images do contain all the information on the satellite’s line of sight, registration of this multi modal set of images is a way to retrieve the satellite’s orientation. This has been suggested in [5,4], but usually this image registration problem is ill-posed as all images are warped and as it contains a deconvolution step. In this case low frequency content is hard to retrieve using fast deterministic methods.

The satellite usually takes several positioning sensors on board such as star trackers or gyros to rectify its attitude. As such sensors need to be very robust to endure space environment, they are costly and not as accurate as common positioning sensors we could use on earth. Whereas the sampling rate of 1-D images is over 700 Hz, inertial sensors sampling rate is usually lower than 16 Hz. Thus they can only provide a low frequency information on the attitude variations. High dynamic perturbation linked to the engines of the satellite cannot be recorded, and may not be rectified on images. This is a major drawback of methods which solely rely on positioning sensors to estimate attitude [6].

In this paper, we propose to fuse image and star tracker informations to provide a fine estimate of the attitude variation; the star tracker provides the low

¹ Usual name for the orientation of the air and space vehicle in flight dynamics science defined by the yaw, the roll, and the pitch.

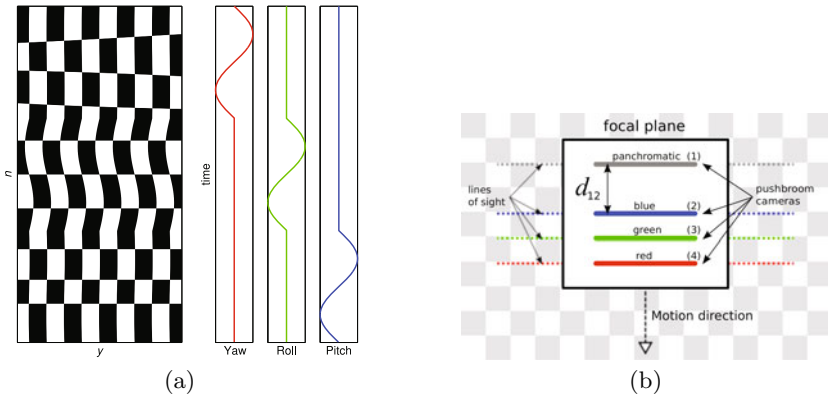


Fig. 2. (a) Example of warps in a regular checkerboard when the pushbroom camera is tilting around its 3 rotation axes. (b) Standard focal plane geometry of observation satellite with 4 pushbroom cameras: panchromatic, blue, green and red (respectively enumerated as 1,2,3 and 4). Let s being the speed motion, what is seen by the camera 2 at time n will be seen by the camera 1 at time $n + \frac{d_{12}}{s}$.

frequency information whereas the image content provides the high frequency information. We start by building a motion model which describes how the images are warped when the attitude is varying. The image and sensor fusion is presented in a Bayesian setting; we use a polynomial model to extract attitude variations from the noisy measurements of the star tracker, and a direct method to match the images and retrieve the attitude variations. We finally show the performance of our algorithm on three satellite datasets.

2 Motion Model

Two pushbroom models were already proposed; one for 3d reconstruction [2] and the other one for calibration [3]. Both assume that the pushbroom camera orientation is fixed over time, consequently none of them can easily describe the warps on the acquired image when the attitude is varying. In this section, we build a warp model depending on the attitude. This model is adapted to the focal plane geometry, and will be further used in the fusion procedure. We assume that the satellite is moving straight along the X axis at a constant speed s , and that the observed scene is very far from the camera. Let $[X_i(n), Y_i, Z_i]^T$ be the position at discrete time n of a pushbroom sensor i ; according to the figure 1, we have the following relationship between camera 1 and camera 2 positions:

$$\begin{bmatrix} X_1(n) \\ Y_1 \\ Z_1 \end{bmatrix} = \begin{bmatrix} X_2(n) - d_{12} \\ Y_2 \\ Z_2 \end{bmatrix} \tag{1}$$

All the cameras share the same known absolute position over time. If the attitude is varying in this case, image 1 and image 2 are the same up to an homography

transformation on each line given by rotation angles. This is similar to the rotational panoramas as described in [7], except that we are in a 1-D case. This is allowed by the specific geometry of the focal plane were the pushbroom cameras are set in parallel. Thereafter, we will denote by $\theta(n) = [\gamma(n), \lambda(n), \phi(n)]^T$ the attitude vector at time n of the focal plane, where its components are respectively the yaw, the roll and the pitch at time n . Let $[0, y_i, 1]^T$ be the pixel coordinates in the image plane for camera i (notice that $x_i = 0$ as we are in a 1-D case). If we take the case of two pushbroom cameras being at the same position and taking a 1-D image up to rotations angles, the mapping equation between images of camera 1 and 2 will be given by:

$$K^{-1}R(\theta(n)) \begin{bmatrix} 0 \\ y_1 \\ 1 \end{bmatrix} \sim K^{-1}R(\theta(n - \tau_{12})) \begin{bmatrix} 0 \\ y_2 \\ 1 \end{bmatrix} \tag{2}$$

where $K = \text{diags}(f, f, 1)$ is the calibration matrix, R the rotation matrix depending on the attitude $\theta(n)$ and $\tau_{12} = \frac{d_{12}}{s}$ with s being the speed of the satellite. We now need to express R ; let assume that rotations are small enough to be linearised to the first order. This is reasonable as the attitude variations we could expect are lower than 1 milli-radian. This gives us the following relation using Euler angles (we temporally drop off n for clarity):

$$\mathbf{R} \simeq \underbrace{\begin{bmatrix} 1 & -\gamma & 0 \\ \gamma & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{Yaw}} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -\lambda \\ 0 & \lambda & 1 \end{bmatrix}}_{\text{Roll}} \underbrace{\begin{bmatrix} 1 & 0 & \phi \\ 0 & 1 & 0 \\ -\phi & 0 & 1 \end{bmatrix}}_{\text{Pitch}} = \begin{bmatrix} 1 + \gamma\lambda\phi & -\gamma - \phi + \gamma\lambda \\ \gamma - \lambda\phi & 1 - \gamma\phi - \lambda \\ \phi & \lambda & 1 \end{bmatrix} \simeq \begin{bmatrix} 1 & -\gamma & -\phi \\ \gamma & 1 & -\lambda \\ \phi & \lambda & 1 \end{bmatrix} \tag{3}$$

If we notice that registering two rotated images is equivalent to retrieving the difference between the rotation angles, we can express equation (2) as:

$$K^{-1} \begin{bmatrix} 0 \\ y_1 \\ 1 \end{bmatrix} \sim K^{-1}R(\theta(n) - \theta(n - \tau_{12})) \begin{bmatrix} 0 \\ y_2 \\ 1 \end{bmatrix} \tag{4}$$

Considering now the 2-D image mapping, we can notice from equation (1) that without any rotations of the cameras: $x_1 = x_2 - d_{12}$. Expanding relation (4) gives us the final motion model for the specific case where pushbroom sensors are aligned in a same focal plane:

$$\begin{aligned} x_1 &= x_2 - d_{12} - \frac{f^{-1}}{(\lambda(n) - \lambda(n - \tau_{12}))y_2 + 1} \left((\gamma(n) - \gamma(n - \tau_{12}))y_2 - (\phi(n) - \phi(n - \tau_{12})) \right) \\ y_1 &= \frac{1}{(\lambda(n) - \lambda(n - \tau_{12}))y_2 + 1} \left(y_2 - (\lambda(n) - \lambda(n - \tau_{12})) \right) \end{aligned} \tag{5}$$

Thereafter, we will call $\mathbf{y}_i = [x_i, y_i]^T$ the vector of pixel coordinates in image i , and $W : \mathcal{S}, \Theta \rightarrow \mathcal{S}$ the warp function given by equation (5) which maps pixel from one image to another in the pixel set \mathcal{S} and attitude set Θ .

3 Data Fusion

The space context is particularly suited to the data fusion context as the satellite is carrying several imaging and positioning sensors. Many articles have already proposed image fusion of panchromatic image with other spectral channels to increase resolution and SNR of acquired images [8]. For attitude control and retrieval, works were also done on the fusion of positioning sensors [6], but to our knowledge, no work was proposed for the fusion of image and sensor to retrieve a fine estimate of the satellite's attitude.

So as to set the problem, we use the following notations: $\mathbf{I} = \{I_p, I_r, I_g, I_b\}$ is the set of four multi-modal images of size (N_i, M) , respectively the panchromatic, the red, the green and blue channel. N_i is the number of time samples in the acquisition process, and M the size of each pushbroom sensor. \mathbf{s} is the $(3N_s \times 1)$ vector of star tracker measurements, N_s being the number of sample acquired by the sensor. We call $\boldsymbol{\theta}$ the $(3N_i \times 1)$ vector that gathers all attitudes for all time instants in an image acquisition. We denote by \mathbf{a} the vector of model parameters for the high frequencies, and \mathbf{b} the vector of model parameters for the low frequency part.

3.1 Bayesian Formulation

The Bayesian formulation allows us to aggregate exogenous informations on a phenomenon to compute its likelihood. In the context of sensors fusion, [9] clearly describes advantages of such an approach where given its noise model, each sensor can be added to the posterior probability, as well as some prior knowledge on the unknown.

Given acquired images and star tracker measurements, we want to infer the attitude $\boldsymbol{\theta}$ such that it follows a model parameterised by \mathbf{a} and \mathbf{b} . We have the following relation:

$$(\hat{\boldsymbol{\theta}}, \hat{\mathbf{a}}, \hat{\mathbf{b}}) = \underset{\boldsymbol{\theta}, \mathbf{a}, \mathbf{b}}{\operatorname{argmax}} p(\boldsymbol{\theta}, \mathbf{a}, \mathbf{b}, \mathbf{I}, \mathbf{s}) \quad (6)$$

$$= \underset{\boldsymbol{\theta}, \mathbf{a}, \mathbf{b}}{\operatorname{argmax}} p(\boldsymbol{\theta}, \mathbf{a}, \mathbf{b} | \mathbf{I}, \mathbf{s}) \quad (7)$$

To expand the previous expression, we will make the following hypothesis: knowing the attitude is enough to describe the likelihood of images ($p(\mathbf{I} | \boldsymbol{\theta}, \mathbf{s}, \mathbf{a}, \mathbf{b}) = p(\mathbf{I} | \boldsymbol{\theta})$), only the star tracker can provide information on \mathbf{b} and we assume a uniform prior on model parameters \mathbf{a} and \mathbf{b} . This yields the following expression:

$$(\hat{\boldsymbol{\theta}}, \hat{\mathbf{a}}, \hat{\mathbf{b}}) = \underset{\boldsymbol{\theta}, \mathbf{a}, \mathbf{b}}{\operatorname{argmax}} p(\mathbf{I} | \boldsymbol{\theta}) p(\boldsymbol{\theta} | \mathbf{a}, \mathbf{b}) p(\mathbf{b} | \mathbf{s}) \quad (8)$$

3.2 Prior Model $p(\boldsymbol{\theta} | \mathbf{a}, \mathbf{b})$

As previously explained, attitude variation of the satellite is due to space turbulences in its low frequency part, and dynamic disturbances of the satellite's engines in its high frequency part. Nonetheless, it is a smooth process which cannot undergo fast motion with discontinuities even in high frequencies. The

engines correspond to a vibratory process which is well described by autoregressive models [10]. Such prior was already successfully adopted in [4]. To complete the model, we choose to approximate low frequencies with a simple polynomial model. So as to avoid redundancy in the equations, we will use α as a subscript on variables to denote either the yaw, the pitch and the roll. This yields the following prior:

$$\theta_\alpha(n) = \underbrace{\sum_{i=1}^{p_\alpha} \theta_\alpha(t-i)a_{i,\alpha}}_{\text{high freq: AR model}} + \underbrace{\sum_{j=1}^{d_\alpha} b_{j,\alpha}t^{j-1}}_{\text{low freq: polynomial model}} + \epsilon(n) \quad (9)$$

where p_α is the order of the autoregressive model, d_α the degree of the polynomial and $\epsilon(n)$ a i.i.d. zero mean Gaussian noise. This equation is linear and the log likelihood of the prior can be written in a matrix form:

$$\log(p(\boldsymbol{\theta}|\mathbf{a}, \mathbf{b})) \propto \|M_{\mathbf{a}}\boldsymbol{\theta}\|^2 + \|\boldsymbol{\theta} - M_t\mathbf{b}\|^2 + \text{cst} \quad (10)$$

where $\|\cdot\|^2$ stands for the l-2 norm. Let $P = p_{\text{yaw}} + p_{\text{pitch}} + p_{\text{roll}}$ and $D = d_{\text{yaw}} + d_{\text{pitch}} + d_{\text{roll}}$, then \mathbf{a} is a $(P \times 1)$ vector, \mathbf{b} a $(D \times 1)$ vector. M_t is a $(3N_i, D)$ matrix which combines linearly with the polynomial coefficients and $M_{\mathbf{a}}$ a $(3N_i - P + 1 \times 3N_i)$ matrix which combines linearly the attitude vector. The constant term accounts for the normalizing factor of the Gaussian p.d.f.

3.3 Star Tracker Term $p(\mathbf{b}|\mathbf{s})$

The star tracker is a positioning sensor which gives an absolute value of the satellite’s attitude. This optical device is looking at stars and tries to register its images with known maps of stars. Its design is strong enough to endure space environment. As a consequence, it can only provide low frequency measurements (below 16Hz) contaminated with a specific colored noise. It is sensitive to very low frequency drift, but to the scale of an image acquisition of a few seconds, a zero mean Gaussian noise is a quite good assumption.

We choose to infer the low frequency parameters of the prior model from the star tracker measurements. In this case, the log likelihood yields:

$$\log(p(\mathbf{b}|\mathbf{s})) \propto \|\mathbf{s} - M_{t_s}\mathbf{b}\|^2 + \text{cst} \quad (11)$$

where M_{t_s} is a $(3N_s \times D)$ matrix which combines linearly with the polynomial coefficients \mathbf{b} .

3.4 Image Data Term $p(\mathbf{I}|\boldsymbol{\theta})$

The panchromatic camera is capturing all wavelengths of visible light, as opposed to the others cameras which record only specific wavelengths ranges (red, blue, and green). To overcome the multi-modal matching problem, we will assume that the panchromatic camera is a linear combination of the other spectral channels. This is expressed by coefficients c_r , c_b and c_g in the following equation:

$$I_p(\mathbf{y}_p) - \left[c_r I_r(W(\mathbf{y}_r; \theta(t) - \theta(t - \tau_{rp}))) + c_b I_b(W(\mathbf{y}_b; \theta(t) - \theta(t - \tau_{bp}))) + c_g I_g(W(\mathbf{y}_g; \theta(t) - \theta(t - \tau_{gp}))) \right] \sim \mathcal{N}(0, \sigma_i^2) \quad (12)$$

where σ_i^2 is the variance of a zero mean Gaussian noise i.i.d. over all pixels of images. The likelihood of the images given the attitude is:

$$\log(p(\mathbf{I}|\boldsymbol{\theta})) \propto \|I_p(\mathbf{y}_p) - (c_r I_r(W(\mathbf{y}_r; M_{k,pr}\boldsymbol{\theta})) + c_g I_g(W(\mathbf{y}_g; M_{k,pg}\boldsymbol{\theta})) + c_b I_r(W(\mathbf{y}_b; M_{k,pb}\boldsymbol{\theta})))\|^2 + \text{cst} \quad (13)$$

where the M_k s are matrices which differentiate $\boldsymbol{\theta}$ depending on the time shift τ . Equation 13 is the expression of a pixel-based registration method [7,11,12,13]. In the context of sub pixel displacement, this method is well suited to estimate the parameters of the warp. This equation can be minimized with any gradient descent method.

3.5 Algorithm

In order to maximize the likelihood of $(\hat{\boldsymbol{\theta}}, \hat{\mathbf{a}}, \hat{\mathbf{b}})$ on equation (8), one could use a fully Bayesian procedure as described in [9]. Unfortunately, MCMC based methods are not suited to the amount of data we need to process, as the convergence rate is too slow as compared to our time constraints (almost real-time) to get an acceptable result. We choose to rely on a multi step algorithm which is sub optimal compared to an MCMC method but yet yields good results:

- compute \mathbf{b} using equation (11):

$$\hat{\mathbf{b}} = \underset{\mathbf{b}}{\text{argmin}} \|\mathbf{s} - M_t \mathbf{b}\|^2$$
- infer $\boldsymbol{\theta}$ from equations (13) and (10) where \mathbf{a} is set to $\mathbf{0}$:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\text{argmin}} \|I_p(\mathbf{y}_p) - (c_r I_r(W(\mathbf{y}_r; M_{k,pr}\boldsymbol{\theta})) + c_g I_g(W(\mathbf{y}_g; M_{k,pg}\boldsymbol{\theta})) + c_b I_r(W(\mathbf{y}_b; M_{k,pb}\boldsymbol{\theta})))\|^2 + \lambda \|\boldsymbol{\theta} - M_t \mathbf{b}\|^2$$

where λ is a trade off scalar parameter between the likelihood and the prior term.
- compute \mathbf{a} from $\boldsymbol{\theta}$ using Yuke-Walker equations [14,4]
- minimize the following equation until convergence:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\text{argmin}} \|I_p(\mathbf{y}_p) - (c_r I_r(W(\mathbf{y}_r; M_{k,pr}\boldsymbol{\theta})) + c_g I_g(W(\mathbf{y}_g; M_{k,pg}\boldsymbol{\theta})) + c_b I_r(W(\mathbf{y}_b; M_{k,pb}\boldsymbol{\theta})))\|^2 + \lambda (\|M_a \boldsymbol{\theta}\|^2 + \|\boldsymbol{\theta} - M_t \mathbf{b}\|^2)$$

We use a cross validation procedure to select both the regularization parameter λ and the polynomial order for the prior model on low frequencies. Radiometric coefficients c_r , c_b and c_g are estimated on images before registration with a standard least square procedure.

4 Experimental Results

This section presents results of our algorithm on 3 satellite datasets; they were simulated by EADS Astrium so that the ground truth is available.

The simulation process aims at reproducing real life acquisition conditions by taking into account measurement noise for each sensor, ground elevation, radiometric distortions and mechanical disturbance of the satellite.

Each dataset is composed of 4 images (panchromatic, blue, green and red) of size (2564×900) pixels, where 900 is the size of the pushbroom sensor. The sampling rate of 1-D images is 770Hz, whereas the sampling rate of the star tracker is 16Hz. For all experiments we use a Matlab implementation on a Core2 duo at 3GHz with 3.8GiB. Our algorithm converges in less than 200 seconds.

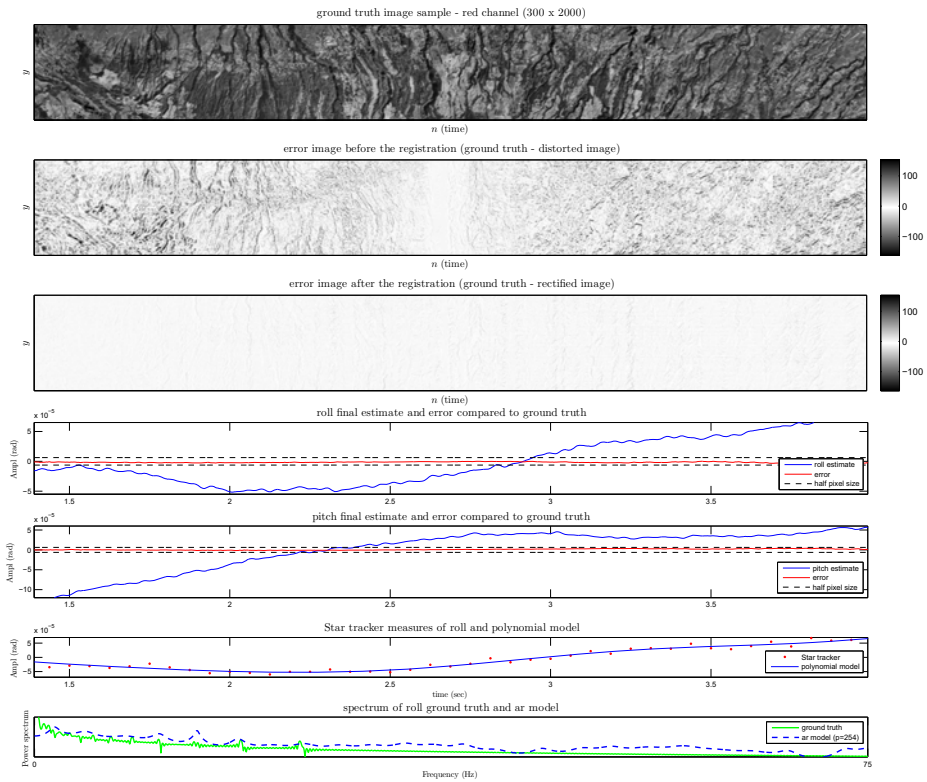


Fig. 3. Dataset 1 results: the top figure is a (300×2000) image patch of the red camera. Below, the following two figures show intensity differences between the real image and the acquired image before and after registration. Following graphs draw attitude variation estimate in radian for the roll and the pitch in blue and the error compared to ground truth in red. The last two figures show the star tracker measurements for the roll in red dots as well as the estimated polynomial model in blue, and the spectrum of the roll ground truth compared to the spectrum of the autoregressive prior.

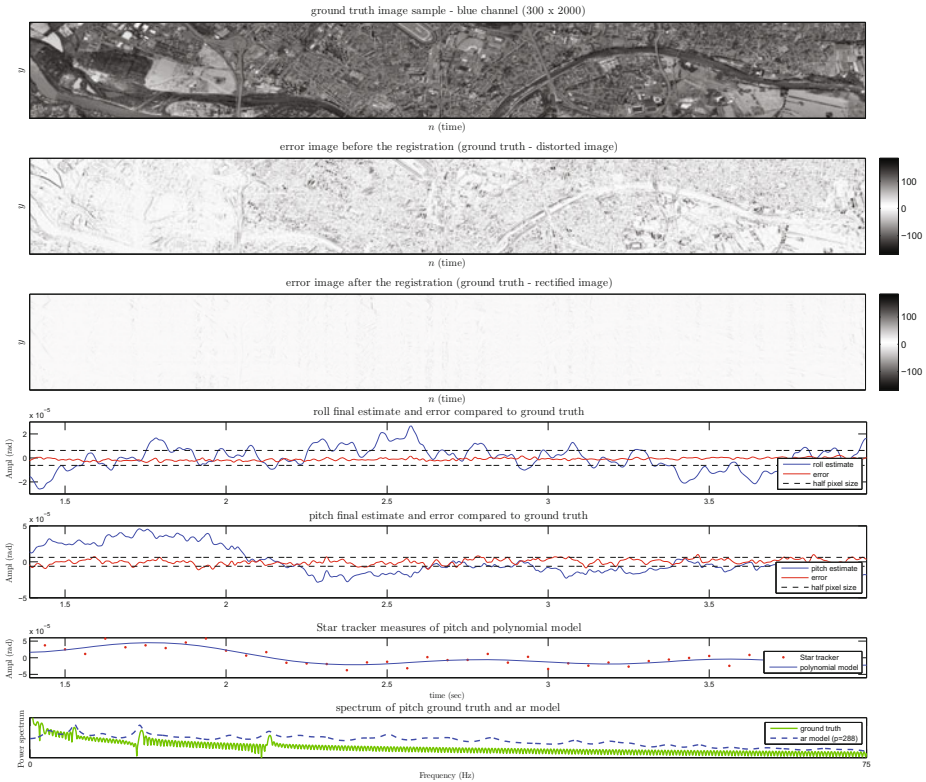


Fig. 4. Dataset 2 results: as for the dataset 1, figures are respectively the ground truth image, the error before and after registration, the pitch and the roll estimates with the error, the startracker measurements and the estimated polynomial model, and the pitch ground truth spectrum compared to the autoregressive model spectrum

All the figures present a (300×2000) patch of the observed scene on the top. Below, two pictures show the error images before and after the registration. We define them as the difference between the ground truth image and the warped image during the acquisition process. The following graphs are respectively the roll and the pitch estimates in blue curves with their errors in red compared to the real attitude, the estimated polynomial model in blue and the star tracker measurements in red dots (plotted for one of the rotation angles), and finally the spectrum of the real attitude in green compared to the autoregressive prior spectrum in dashed blue line (also plotted for one of the rotation angles).

In all the experiments one should notice how the autoregressive prior is trying to fit the high frequencies of the attitude while the polynomial model is providing low frequency information on the estimate.

The first dataset (figure 3) is a challenging case where the attitude variations contain very low frequencies and a displacement of several pixels. The accuracy we obtained either for the roll and the pitch estimate is below $\frac{8}{100}$ in pixel, which is a good score in such ill-posed problem.

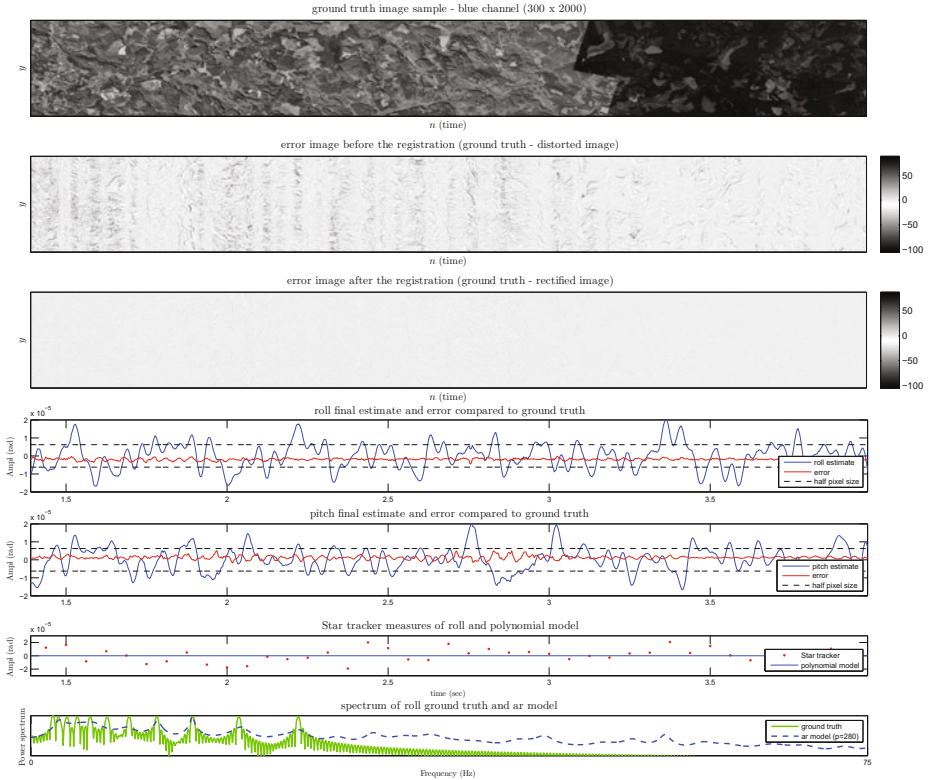


Fig. 5. Dataset 3 results: as for previous dataset, figures are respectively the ground truth image, the error before and after registration, the pitch and the roll estimates with the error, the startracker measurements and the estimated polynomial model, and the pitch ground truth spectrum compared to the autoregressive model spectrum

The second dataset (figure 4) is the only one to be mono-modal in all the acquired images; all the cameras capture the same light spectrum. It has lower performances in the pitch estimate (around $\frac{1}{4}$ in pixel accuracy), and we may link this to our radiometric model with is not suited anymore. We did not conduct enough experiments yet to evaluate the advantages of our radiometric model, but we noticed that the image registration was converging faster. This linear relationship between panchromatic and RGB modalities has seemingly not been exploited yet in satellite image processing and may leads to interesting solutions in image fusion, resampling and demosaicing.

The third dataset (figure 5) is a tricky case where the real attitude has no low frequency component. As we can see, the polynomial model is not misled by the star tracker measurements as the degree of the polynomial is zero. The autoregressive prior is fitting most of the high frequencies and we achieved an accuracy of $\frac{1}{10}$ in pixel unit.

The motion model we described in section 2 is accurate enough in our experiments, though it assumes that the observed scene is planar. Such assumption is

weakened if the scene has a strong relief. In such a case we need to use Digital Elevation Model of the earth to compute numerical derivatives of the warp [4] but this is computationally expensive.

5 Conclusion

In this paper, we have presented a data fusion algorithm to get a fine estimate of the attitude variations of a satellite. Up to our knowledge, our method is the only one that proposes a fusion of pushbroom image content and positioning sensor data. The results we got are promising and we believe that the model we selected is well suited to the attitude estimation for observation satellite. The simulated we used is a first step to see how our method performs as the ground truth is available; we are looking forward to process a large set of real satellite data to validate our algorithm.

There are still room for improvement on the warp model which could take into account non planar scene. The radiometric model we choose also need to be finely evaluated, but it may open nice perspective on satellite image processing. Finally the global algorithm may be improved to yield a better estimate using hybrid MCMC methods with gradient descent.

Acknowledgement. This work was funded by EADS Astrium (European aerospace company and satellite manufacturer).

References

1. Petrie, G.: Airborne pushbroom line scanners: An alternative to digital frame scanners. *Geoinformatics* 8, 50–57 (2005)
2. Gupta, R., Hartley, R.I.: Linear pushbroom cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19, 963–975 (1997)
3. Drareni, J., Sturm, P., Roy, S.: Plane-Based Calibration for Linear Cameras. In: *The 8th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras - OMNIVIS* (2008)
4. Perrier, R., Arnaud, E., Sturm, P., Ortner, M.: Estimating satellite attitude from pushbroom sensors. In: *CVPR* (2010)
5. de Lussy, F., Greslou, D., Gross Colzy, L.: Process line for geometrical image correction of disruptive microvibrations. In: *International Society for Photogrammetry and Remote Sensing*, pp. 27–35 (2008)
6. Poli, D.: General model for airborne and spaceborne linear array sensors. *International Archives of Photogrammetry and Remote Sensing* 34 (2002)
7. Szeliski, R.: Image alignment and stitching: a tutorial. *Found. Trends. Comput. Graph. Vis.* 2, 1–104 (2006)
8. Blum, R., Liu, Z.: *Multi-sensor image fusion and its applications*. CRC, Boca Raton (2006)
9. Punska, O.: *Bayesian approaches to multi-sensor data fusion*. Cambridge University, Cambridge (1999)
10. Doebbling, S., Farrar, C., Prime, M., Shevitz, D.: Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: A literature review. Technical report, Los Alamos National Lab. (1996)

11. Baker, S., Matthews, I.: Lucas-Kanade 20 years on: A unifying framework. *International Journal of Computer Vision* 56, 221–255 (2004)
12. Irani, M., Anandan, P.: About direct methods. In: *Proceedings of the International Workshop on Vision Algorithms*, pp. 267–277 (1999)
13. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI 1981)*, pp. 674–679 (1981)
14. Makhoul, J.: Linear prediction: A tutorial review. *Proceedings of the IEEE* 63, 561–580 (1975)

Image Segmentation Fusion Using General Ensemble Clustering Methods

Lucas Franek¹, Daniel Duarte Abdala¹, Sandro Vega-Pons², and Xiaoyi Jiang¹

¹ Department of Mathematics and Computer Science,
University of Münster, Germany

² Advanced Technologies Application Center (CENATAV), Havana, Cuba
{lucas.franek,xjiang}@uni-muenster.de, danielabdala@gmail.com,
sv.pons@gmail.com

Abstract. A new framework for adapting common ensemble clustering methods to solve the image segmentation combination problem is presented. The framework is applied to the parameter selection problem in image segmentation and compared with supervised parameter learning. We quantitatively evaluate 9 ensemble clustering methods requiring a known number of clusters and 4 with adaptive estimation of the number of clusters. Experimental results explore the capabilities of the proposed framework. It is shown that the ensemble clustering approach yields results close to the supervised learning, but without any ground truth information.

1 Introduction

Image segmentation is the first step and also one of the most critical tasks in image analysis. In order to deal with the great variability of features encountered in different images specific segmentation methods have been designed for different types of images, including medical [1], range [2], and outdoor images [3] among many other examples. Many of these image segmentation methods also do require that appropriate parameters have to be selected in order to achieve a good segmentation result. There exists no general unsupervised method for effectively selecting the best parameters. Thus, usually researchers use supervised parameter learning to estimate a fixed parameter setting [3].

Recently, a new direction in image segmentation has been taken in order to deal with this general problem. Instead of selecting one optimal parameter setting it was proposed to combine several different segmentations received by different parameter settings or different segmentation algorithms into a final consensus segmentation. This approach is known as image segmentation combination [4]. Some combination methods can be found in the literature specifically designed to deal with the image segmentation combination problem [4, 5, 6]. They take into account details such the size of the datasets and well structured pattern's lattice.

¹ In some papers, the terms image fusion and image merging are used. We prefer to use the term image segmentation combination since the other terms can also appear in different contexts.

This work addresses the parameter selection problem by applying general ensemble clustering methods in order to produce a consensus segmentation. This approach is motivated by an inherent relation of both tasks: Ensemble clustering and segmentation combination aim to combine a set of solutions into a final consensus solution. Recently, there has been some work done applying general ensemble clustering methods to the image segmentation combination problem [7,8,9]. The authors of these works claim to improve resulting segmentations by this kind of combination. However, in these works ensemble clustering methods mostly are used in combination with other heuristics and quantitative experimental results are not provided or limited. Our work builds on the previously cited works and provides a broad experimental study. The main contribution of our work consists of applying and comparing a broad variety of representative and widely used ensemble clustering methods to the segmentation combination problem. Furthermore we compare this approach to the supervised parameter learning approach. It will be examined if comparable or even superior results are received without knowing ground truth. By this way we aim to justify the usefulness of ensemble clustering methods in the context of segmentation combination.

In order to make image datasets processable by such general ensemble clustering combination methods, some pre- and post-processing steps are required. A framework is proposed allowing virtually any general ensemble clustering method to be used in such context.

This paper is organized as follows. Section 2 reviews the ensemble clustering methods used in our study. The pre- and post-processing steps which are used in the proposed framework are detailed. Section 3 describes the performed experiments and discriminates the used datasets. In Section 4 experimental results are reported, followed by some conclusions and our final remarks in Section 5.

2 Framework for Segmentation Combination by General Ensemble Clustering Methods

Given a set of segmentations $I = \{S_1, \dots, S_M\}$, the problem of segmentation combination is to combine the segmentations into a consensus segmentation S^* which in some sense optimally represents the ensemble I . The goal of ensemble clustering methods is quite related, as will be explained in the following. For this reason let $X = \{x_1, x_2, \dots, x_N\}$ denote a dataset of N objects x_i . A set of clustering results is a set $\mathbb{P} = \{P_1, P_2, \dots, P_M\}$, where P_i is a partition of X produced by clustering X and M is the number of partitions. We denote the set of all possible partitions of X by \mathbb{P}_X ($\mathbb{P} \subset \mathbb{P}_X$). The goal of ensemble clustering methods is to find a consensus clustering $P^* \in \mathbb{P}_X$, which optimally represents the ensemble \mathbb{P} .

In order to be able to use any existing ensemble clustering method for the task of image segmentation combination the following processing pipeline (Fig. II) is proposed:

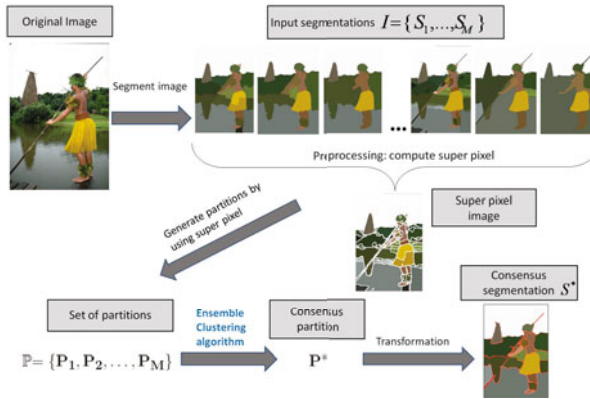


Fig. 1. Processing pipeline: cluster ensemble \mathbb{P} is computed by using super pixels. General clustering combination methods are used to generate a consensus clustering P^* , which is transformed into the final consensus segmentation S^* .

1. Produce M segmentations $I = \{S_1, \dots, S_M\}$ of an image by varying parameters or using different segmentation algorithms.
2. Generate super pixels and eliminate small super pixels to further reduce the number of objects.
3. Compute the set of clusterings \mathbb{P} by using super pixels.
4. Apply a general ensemble clustering method to \mathbb{P} and receive a consensus clustering P^* .
5. Post-processing step: P^* is transformed into a consensus segmentation S^* .

The remainder of this section reviews in detail each one of the used combination methods, the pre-processing step in order to ensure the diminishment of the number of objects as well the necessary post-processing.

2.1 Pre-processing of the Image Segmentation Ensemble

Image based datasets are known to contain a large number of pixels. In dealing with image segmentation combination, this number is further enlarged by the number of the segmentation samples in the ensemble, leading to a considerable workload. Thus, any useful combination method requires some sort of diminishment in the number of objects to be processed.

The proposed pre-processing step in our framework is motivated by the fact that neighboring pixels, which are equally labeled in each segmentation, do not have to be clustered individually by the ensemble clustering algorithm. Thus Singh *et al.* [8] proposed to compute a representative object called super pixel for each such group of pixels. The pixels of the image are divided into non-overlapping subsets of pixels (super pixels) such that for each segmentation of I , pixels in each super pixel are equally labeled. By using super pixels I is now transformed to the set \mathbb{P} , which may be used as input for the ensemble clustering

method. The size of objects in \mathbb{P} is at least the maximum number of segments in the original segmentations $S_i \in I$ and at most the number of pixels in the image, which is very unlikely. However, because some segmentation algorithms are known to be inaccurate at boundaries in some regions there may be a large number of very small super pixels. We decided to eliminate these super pixels. Therefore, they have to be handled in the post-processing step.

2.2 Ensemble Clustering Methods

This section reviews the ensemble clustering methods used in our evaluation.

BOK (Best of K): The idea behind Best of K is to select the best or most representative partition among all partitions in \mathbb{P} . This is achieved by selecting iteratively each partition in \mathbb{P} and computing the sum of distances (SoD) between the selected partition and the remaining ones in \mathbb{P} .

$$SoD(P) = \sum_{i=1}^M d(P_i, P) \quad (1)$$

The partition $P \in \mathbb{P}$ with smallest SoD value is selected as consensus partition.

BOEM: The Best One Element Moves [10] starts with an initial consensus clustering partition. We can select any method such as BOK or EAC-SL/AL (which is explained in the following) as initial result. The algorithm follows by interactively testing each possible label for each object, retaining the label that decreases the SoD.

EAC SL/AL: The method proposed in [11] explores the idea of evidence accumulation by combining M partitions generated over the same dataset into a co-association matrix. Each cell in this matrix has the value $C(i, j) = \frac{m_{i,j}}{M}$, where $m_{i,j}$ refers to how many times the pair (i, j) of objects occurs in the same cluster among the M clusterings. This matrix can be viewed as a new similarity measure between the set of objects X . The more frequent objects x_i and x_j appear in the same clusters, the more similar they are. Using the co-association matrix C as the similarity measure between objects, the consensus partition is obtained by applying a hierarchical agglomerative clustering algorithm. In the experiments we used the single-link and average-link algorithms.

RW: The general idea that motivates the random walker method [5] is to create a graph representation of the dataset and then apply a random walker based heuristic to infer the consensual partition. It can be divided in 3 parts: a) graph generation; b) seed region generation; and c) ensemble combination. In the graph generation the data is pre-processed in order to create a graph representation $G(V, E, W)$. For the vertex set V a vertex corresponding to each object is defined. To generate E the algorithm iterates over all vertices and edge weights are computed. A weight $w_{i,j}$ indicates how probably the two objects x_i and x_j belong to the same cluster. Clearly, this can be guided by counting the number $m_{i,j}$ of initial partitions in the same manner as described in **EAC SL/AL**.

Seed regions are computed from the resulting graph (for details please refer to [5]). The method allows both automatic selection of the optimal number of seed regions and the definition of a fixed number of target clusters. The ensemble combination uses the graph G constructed from the initial partitions and K seed regions, over which the random walker algorithm [12] is applied to compute the consensus segmentation.

Hypergraph based methods: Strehl and Ghosh [13] proposed three heuristics based on hypergraph partitioning: CSPA, HGPA and MCLA. The three heuristics represent \mathbb{P} as a hypergraph, whereas each partition is represented by a hyperedge.

Cluster-based Similarity Partitioning Algorithm (CSPA). In this method, an $N \times N$ similarity matrix is defined from the hypergraph. This can be viewed as the adjacency matrix of a fully connected graph, where the nodes are the elements of the set X and an edge between two objects has an associated weight equal to the number of times the objects are in the same cluster. Then, the graph partitioning algorithm METIS [14] is used to obtain the consensus partition.

HyperGraphs Partitioning Algorithm (HGPA). This method partitions the hypergraph directly by eliminating the minimal number of hyperedges. It is considered that all hyperedges have the same weight, and it is searched by cutting the minimum possible number of hyperedges that partitions the hypergraph in k connected components of approximately the same dimension. For the implementation the hypergraph partitioning package HMETIS [15] is used.

Meta-CLustering Algorithm (MCLA). In this method the similarity between two clusters is defined in terms of the amount of objects grouped in both, using the Jaccard index. Then, a similarity matrix between clusters is formed which represents the adjacency matrix of the graph. It is built by considering the clusters as nodes and assigning a weight to the edge between two nodes, whereas the weight represents the similarity between the clusters. This graph is partitioned using the METIS [14] algorithm and the obtained clusters are called meta-clusters. Finally, to find the consensus partition each object is assigned to its most associated meta-cluster.

Information theory based methods: Topchy *et al.* [16] introduced the *Quadratic Mutual Information (QMI)* based algorithm. In this method, the *category utility function* U [17] is used as a similarity measure between two partitions. In this case, the category utility function $U(P_i, P_j)$ can be interpreted as the difference between the prediction of the clusters of a partition P_i both with the knowledge of the partition P_j and without it. This way, the better agreement between the two partitions, the higher values of the category utility function we shall have. Hence, the consensus partition could be defined by using U as a similarity measure between partitions:

$$P^* = \arg \max_{P \in \mathbb{P}_X} \sum_{i=1}^M U(P, P_i) \quad (2)$$

This problem is equivalent to the minimization of the square-error clustering criterion if the number of clusters k is known for the consensus partition. This way the solution of the problem (2) is approached in the following way. First, for each object the values of new features are computed using the information in the cluster ensemble. After that, the final partition is obtained by applying the k-Means algorithm on the new data.

Kernel based methods: Vega-Pons *et al.* [18] proposed the *Weighted Partition Consensus via Kernels* (WPCK) algorithm. In this method, the consensus partition is defined as:

$$P^* = \arg \max_{P \in \mathbb{P}_X} \sum_{i=1}^M \omega_i \cdot \hat{k}(P, P_i)$$

where ω_i is a weight associated to partition P_i and \hat{k} is a similarity measure between partitions, which is a kernel function. The weight values ω_i are usually computed in a step before the combination, where the relevance of each partitions is estimated. However, in this paper, we do not consider the weights because their computation needs the use of the original data. Then, for us $\omega_i = 1, \forall i = 1, \dots, M$. The kernel property of \hat{k} allows mapping this problem into a Hilbert Space \mathcal{H} , where an exact solution can be easily obtained. Given the solution in \mathcal{H} the pre-image problem could be solved, i.e., finding the partition in \mathbb{P}_X which corresponds with the solution in \mathcal{H} . This is usually a hard optimization problem that could not have an exact solution. The simulated annealing meta-heuristic was used to obtain an approximated solution avoiding the convergence to local minima. In this algorithm, the specification of the number of clusters in the final partition is not necessary. However, it can be modified to work with a fixed number of clusters k in the final partition. This can be done by applying the simulated annealing but only considering as new states in the process, partitions with k clusters.

Clustering based on semidefinite programming: SDP [8] is motivated by the observation that pairwise similarity values between objects as used in [13] do not provide sufficient information for ensemble clustering algorithms. Therefore, the authors propose to the solutions obtained by individual clustering results by a multidimensional string. In the first step a so-called A-string is computed for every data element, which encodes the information from the individual clustering results. The ensemble clustering problem reduces to a form of string clustering problem where the objective is to cluster similar strings to the same cluster. For this reason the authors first formulate a non-linear objective function which is transformed into a 0-1 semidefinite program (SDP) using a convexification technique. This program is then relaxed to a polynomial time solvable SDP.

2.3 Post-processing

After applying a general clustering combination method to \mathbb{P} a consensus clustering P^* is received. By using super pixels P^* is transformed into a consensus segmentation S^* . Because of eliminating small super pixels before computing

\mathbb{P} there will be some unlabeled pixels. These pixels are simply merged to the neighboring region with the smallest color difference.

3 Experiments

In this section we describe the generated datasets used to evaluate our framework. The experiments and evaluation measures are detailed.

3.1 Datasets

We used the color images from the Berkeley dataset [19] to make the experimental comparison of the algorithms described in Section 2.2. The Berkeley dataset is widely used for image segmentation evaluation and it is composed of 300 natural images of size 481×321 . For each image in the dataset, we used 3 state-of-art segmenters to generate 3 ensembles: *TBES ensembles*, *UCM ensembles* and *TBES & UCM ensembles*. Each ensemble is composed of 10 segmentations obtained by varying the parameter values of the segmentation algorithms used to generate the ensemble. *TBES ensembles* were generated with the TBES algorithm [20], which is based on the MDL-principle and has as parameter the quantization level (ϵ). We varied $\epsilon = 40, 70, 100, 130, \dots, 310$ to obtain the 10 segmentations in the ensemble. Furthermore, *UCM ensembles* were generated with a segmenter based on ultrametric contour map (UCM) [21]. Its only parameter is the threshold l , we choose $l = 0.03, 0.11, 0.19, 0.27, 0.35, 0.43, 0.50, 0.58, 0.66, 0.74$. Finally, *TBES & UCM ensembles* were generated by using two different segmenters: TBES and UCM. Five segmentations were obtained with TBES ($\epsilon = 40, 100, 160, 220, 280$) and the others with UCM ($l = 0.03, 0.19, 0.35, 0.50, 0.66$).

3.2 Combination by Ensemble Clustering vs. Supervised Learning

Considering the parameter selection problem in image segmentation we want to provide a general insight into the capability of general ensemble clustering methods. We want to explore how powerful such methods are in the context of segmentation combination. For this reason we proceed as follows:

Combination by ensemble clustering: First for each segmentation ensemble the pre-processing step described in Section 2.1 is applied. Some ensemble clustering algorithms have a parameter k , which specifies the number of regions in the consensus result. This is the case for CSPA, HGPA, MCLA, EAC-SL, EAC-AL and SDP. Thus, for these algorithms for each ensemble k is set equal to the average number of regions of the images of the ensemble. The other algorithms BOK, BOEM, RW and WPCCK do not need any parameter specification. In the experiments, we also used RW and WPCCK with a fixed k value (denoted by RWfixed and WPCCKfixed).

Supervised parameter learning: In order to gain further insight into the power of the framework we decided to apply supervised parameter learning to the same datasets. Therefore, for each dataset we compute the average performance

Table 1. Ensemble clustering results for free parameter k . Ensemble clustering algorithms are applied to each dataset and performance of the consensus segmentation is evaluated. Lower values are better.

		1 - NMI		VI		1 - RI		1 - F-meas.	
Dataset	Method	bestGT	allGT	bestGT	allGT	bestGT	allGT	bestGT	allGT
TBES ensembles	BOK	0.41	0.48	1.34	1.73	0.21	0.28	0.56	0.63
	BOEM	0.35	0.42	1.52	1.82	0.16	0.22	0.45	0.52
	RW	0.49	0.55	1.57	1.97	0.28	0.34	0.58	0.64
	WPCK	0.32	0.39	1.58	1.85	0.15	0.22	0.42	0.49
UCM ensembles	BOK	0.34	0.40	1.90	2.17	0.15	0.21	0.43	0.51
	BOEM	0.41	0.46	2.20	2.44	0.19	0.25	0.49	0.56
	RW	0.43	0.48	1.87	2.15	0.22	0.27	0.50	0.57
	WPCK	0.34	0.40	2.06	2.32	0.15	0.21	0.43	0.51
TBES & UCM ensembles	BOK	0.51	0.56	1.34	1.77	0.29	0.37	0.56	0.63
	BOEM	0.38	0.45	1.58	1.86	0.20	0.25	0.45	0.52
	RW	0.42	0.48	1.32	1.68	0.21	0.28	0.50	0.57
	WPCK	0.31	0.37	1.66	1.92	0.14	0.20	0.40	0.47

measure over all 300 images of Berkeley dataset for each parameter setting. The parameter setting with the largest value is selected as the optimal fixed parameter setting for the corresponding dataset. By this means we may provide a quantitative comparison with the proposed approach.

3.3 Evaluation of Segmentations

In the experiments, we compared the obtained results with the human segmentations (*ground truth*) of each image. We used four well-known measures to evaluate the algorithm results: Normalized Mutual Information (NMI) [13], Variation of Information (VI) [22], Rand Index (RI) [23] and F-measure [19].

NMI, RI and F-measure are similarity measures that take values in the range $[0, 1]$, where 1 means a perfect correspondence between the segmentation and the ground truth. On the other hand, VI is a dissimilarity measure that takes values in $[0, +\infty]$, where 0 means a perfect correspondence between segmentations. In order to show experimental results in a homogeneous way we present a dissimilarity version of the measures NMI, RI and F-measure. Therefore, we compute the values $1 - \mathcal{SM}$, where \mathcal{SM} represents NMI, RI and F-measure respectively, whereas lower measure values mean better correspondence.

4 Results

The Berkeley database provides for every image several ground truth segmentations. Because pairwise ground truth segmentations for the same image can differ for our experiment we decided to handle this problem by evaluating our results using two different strategies in order to get objective results. First, we take for each segmentation the ground truth image which yields the maximum performance value (denoted as “best GT”). Secondly, we take the mean over all performance values received from different ground truths (“all GT”).

Table 2. Ensemble clustering results for fixed parameter k . Ensemble clustering algorithms are applied to each dataset and performance of the consensus segmentation is evaluated. Lower values are better.

Dataset	Method	1 - NMI		VI		1 - RI		1 - F-meas.	
		bestGT	allGT	bestGT	allGT	bestGT	allGT	bestGT	allGT
TBES ensembles	CSPA	0.33	0.39	1.75	1.99	0.14	0.21	0.42	0.49
	EAC_SL	0.33	0.39	1.43	1.71	0.16	0.21	0.42	0.49
	EAC_AL	0.32	0.39	1.51	1.78	0.15	0.21	0.41	0.48
	HGPA	0.32	0.38	1.75	1.98	0.14	0.21	0.42	0.49
	MCLA	0.34	0.41	1.47	1.77	0.16	0.22	0.44	0.51
	QMI	0.33	0.39	1.68	1.93	0.15	0.21	0.44	0.50
	RWfixed	0.41	0.47	1.82	2.08	0.22	0.28	0.49	0.55
	SDP	0.32	0.38	1.91	2.16	0.14	0.21	0.41	0.48
WPCKfixed	0.32	0.39	1.53	1.80	0.15	0.20	0.41	0.48	
UCM ensembles	CSPA	0.34	0.40	1.90	2.17	0.15	0.21	0.43	0.51
	EAC_SL	0.35	0.41	1.89	2.16	0.15	0.23	0.43	0.51
	EAC_AL	0.35	0.41	1.90	2.17	0.15	0.21	0.43	0.51
	HGPA	0.42	0.49	3.67	4.00	0.18	0.27	0.53	0.62
	MCLA	0.36	0.42	1.91	2.18	0.16	0.22	0.44	0.52
	QMI	0.37	0.43	2.26	2.52	0.16	0.24	0.48	0.55
	RW fix k	0.35	0.41	2.06	2.33	0.15	0.21	0.44	0.52
	SDP	0.34	0.40	2.20	2.47	0.14	0.21	0.44	0.52
WPCKfixed	0.34	0.40	1.90	2.17	0.15	0.21	0.43	0.51	
TBES & UCM ensembles	CSPA	0.32	0.38	2.14	2.42	0.14	0.22	0.41	0.48
	EAC_SL	0.29	0.36	1.46	1.74	0.13	0.19	0.35	0.43
	EAC_AL	0.28	0.35	1.59	1.86	0.12	0.19	0.35	0.43
	HGPA	0.34	0.40	2.27	2.56	0.15	0.22	0.43	0.51
	MCLA	0.34	0.40	1.41	1.71	0.17	0.22	0.41	0.49
	QMI	0.31	0.37	1.82	2.08	0.13	0.20	0.39	0.46
	RWfixed	0.30	0.36	1.69	1.97	0.13	0.20	0.37	0.45
	SDP	0.29	0.36	1.72	2.00	0.13	0.19	0.37	0.45
WPCKfixed	0.30	0.36	1.66	1.93	0.13	0.20	0.38	0.45	

Table 1 shows the results for algorithms with free parameter k . For NMI WPCK outperforms the other ensemble clustering algorithms on all datasets and for VI RW is the best for two datasets. For RI and F-measure WPCK is best, whereas the less complex algorithm BOK only for VI yields very good results. Considering the results for fixed k in Table 2 we observe that there is no considerable variability among NMI , RI and F-measure. If NMI , RI and F-measure are considered three algorithms outperform the others slightly: EAC_AL, SDP and WPCK. In contrast, for VI EAC_SL and MCLA yield slightly better results. It is hard to judge why VI prefers these algorithms. Apart from its desirable properties the relevance of VI for image segmentation is unclear and has to be further explored. For two methods (RW and WPCK) the results for fixed and free parameter k can be directly compared. In both cases the results for fixed k are better than the results for free k . However, it must be emphasized that in some situations heuristics for fixing k are insufficient and methods which adaptively select k are preferred.

Table 3. Performance evaluation of supervised learning and average performance of ensembles. Lower values are better.

		1 - NMI		VI		1 - RI		1 - F-meas.	
	Ensembles	bestGT	allGT	bestGT	allGT	bestGT	allGT	bestGT	allGT
Supervised learning	TBES	0.31	0.37	1.34	1.69	0.14	0.20	0.40	0.47
	UCM	0.28	0.35	1.29	1.61	0.11	0.18	0.32	0.41
	TBES & UCM	0.29	0.36	1.29	1.62	0.13	0.19	0.33	0.42
Average ensemble performance	TBES	0.34	0.41	1.53	1.83	0.16	0.22	0.44	0.51
	UCM	0.36	0.42	1.88	2.25	0.17	0.24	0.42	0.51
	TBES & UCM	0.35	0.42	1.53	1.87	0.17	0.24	0.43	0.51

The results for supervised parameter learning are shown in Table 3. Considering the results for fixed k , for the *TBES* and *TBES&UCM* dataset many ensemble clustering methods yield results close to those received by parameter learning. This is especially the case for *EAC_AL*, *SDP* and *WPCKfixed*. For *NMI* even better results are received for *EAC_AL* (*TBES&UCM* dataset).

Our results give raise to the assumption that good segmentation results may be received by using general ensemble clustering methods like *EAC_AL*, *SDP* or *WPCK* *without knowing ground truth*. In this context it must be emphasized that in many application scenarios supervised learning is not applicable because ground truth is not available. Thus, ensemble clustering methods are preferred in scenarios where parameters of segmentation algorithms are unknown.

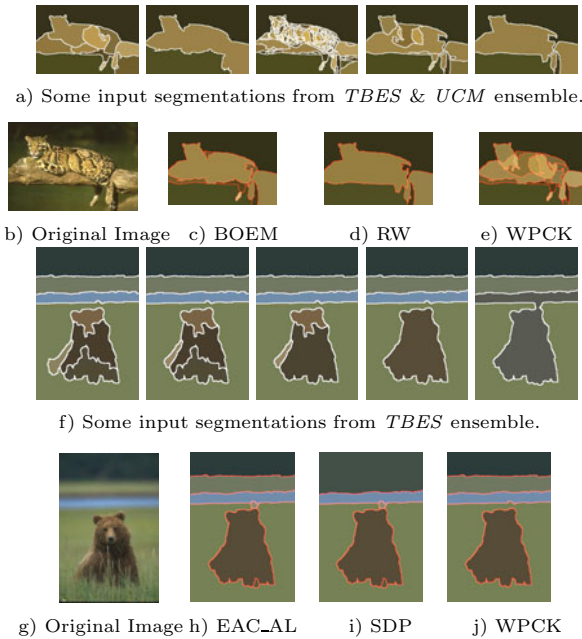


Fig. 2. Consensus segmentation results for free k (c-e), and for fixed k (h-j)

To further illustrate the capability of the methods for each dataset the average ensemble performance AEP is determined which reflects the average quality of the image segmentation ensembles. The AEP is determined by computing the average performance value for each ensemble in a dataset and then averaging over all these values (Table 3). Here we only note that e.g. for the *TBES&UCMensembles* nearby all ensemble clustering algorithms yield better performance values than the average ensemble performance.

Fig. 2 shows some ensemble clustering results for free and fixed k . If k is fixed the ensemble clustering algorithms EAC_AL, SDP and WPCCK perform similar (Fig. 2 h)-j)) as was also seen by analyzing the performance values in Table 2. However, for free k the results may be very different (Fig. 2 c) - e)) which is not surprising. In both cases the input segmentations are nicely combined.

From our experiments we conclude that satisfying segmentation results may be received by using ensemble clustering methods (e.g. EAC_AL). The parameter selection problem can be solved to a certain degree. In this sense our benchmark pointed out some landmarks concerning the combination of segmentations and may be the base for future research. Future work is on how to improve methods like EAC_AL, SDP and WPCCK for the task of segmentation combination.

5 Conclusion

In this work we have proposed a methodology that allows the usage of virtually any ensemble clustering method to address the problem of image segmentation combination. For our knowledge this is the first work that addresses the problem of image segmentation combination from this perspective. The proposed framework deals nicely with the dimensionality problem. A pre-processing step transforms similar neighboring pixels from the segmented images into a single object (super pixel approach). A broad class of general clustering algorithms were applied and compared in the experimental results. The resulting consensus segmentations seem to indicate that indeed smoother results are obtained. By this way results performing as well as the supervised parameter learning are achieved. In this sense the parameter selection problem can be solved to a certain degree. The performed experiments corroborate such observation.

Acknowledgment. Daniel D. Abdala thanks the CNPq, Brazil-Brasilia for granting him a Ph.D. scholarship under the process number 290101-2006-9.

References

1. Suri, J., Setarehdan, S., Singh, S.: Advanced Algorithmic Approaches to Medical Image Segmentation: State-of-the-Art Applications in Cardiology, Neurology, Mammography and Pathology. Springer, Heidelberg (2002)
2. Hoover, A., Jean-baptiste, G., Jiang, X., Flynn, P.J., Bunke, H., Goldgof, D., Bowyer, K., Eggert, D., Fitzgibbon, A., Fisher, R.: An experimental comparison of range image segmentation algorithms. IEEE Trans. on PAMI 18, 673–689 (1996)

3. Unnikrishnan, R., Pantofaru, C., Hebert, M.: Toward objective evaluation of image segmentation algorithms. *IEEE Trans. on PAMI* 29, 929–944 (2007)
4. Cho, K., Meer, P.: Image segmentation from consensus information. *Computer Vision and Image Understanding* 68, 72–89 (1997)
5. Wattuya, P., Rothaus, K., Prañni, J.S., Jiang, X.: A random walker based approach to combining multiple segmentations. In: *Proc. of the 19th Int. Conf. on Pattern Recognition* (2008)
6. Yu, Z., Zhang, S., Wong, H.-S., Zhang, J.: Image segmentation based on cluster ensemble. In: Liu, D., Fei, S., Hou, Z., Zhang, H., Sun, C. (eds.) *ISNN 2007. LNCS*, vol. 4493, pp. 894–903. Springer, Heidelberg (2007)
7. Keuchel, J., Küttel, D.: Efficient combination of probabilistic sampling approximations for robust image segmentation. In: Franke, K., Müller, K.-R., Nikolay, B., Schäfer, R. (eds.) *DAGM 2006. LNCS*, vol. 4174, pp. 41–50. Springer, Heidelberg (2006)
8. Singh, V., Mukherjee, L., Peng, J., Xu, J.: Ensemble clustering using semidefinite programming with applications. *Mach. Learn.* 79, 177–200 (2010)
9. Zhang, X., Jiao, L., Liu, F., Bo, L., Gong, M.: Spectral clustering ensemble applied to texture features for sar image segmentation. *IEEE Transactions on Geoscience and Remote Sensing* 46, 2126–2135 (2008)
10. Goder, A., Filkov, V.: Consensus clustering algorithms: Comparison and refinement. In: *Proceedings of ALENEX*, pp. 109–117 (2008)
11. Fred, A.L., Jain, A.K.: Combining multiple clusterings using evidence accumulation. *IEEE Trans. on PAMI* 27, 835–850 (2005)
12. Grady, L.: Random walks for image segmentation. *IEEE Trans. on PAMI* 28, 1768–1783 (2006)
13. Strehl, A., Ghosh, J.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *J. on Machine Learning Research* 3, 583–617 (2002)
14. Karypis, G., Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal of Scientific Computing* 20, 359–392 (1998)
15. Karypis, G., Aggarwal, R., Kumar, V., Shekhar, S.: Multilevel hypergraph partitioning: application in vlsi domain. In: *Proceedings of the 34th Annual Conference on Design Automation, DAC 1997*, pp. 526–529. ACM, New York (1997)
16. Topchy, A.P., Jain, A.K., Punch, W.F.: Clustering ensembles: Models of consensus and weak partitions. *IEEE Trans. Pattern Anal. Mach. Intell.* 27, 1866–1881 (2005)
17. Gluck, M., Corter, J.: Information, uncertainty, and the utility of categories. In: *Proc. of the Seventh Annual Conference of the Cognitive Science Society*, pp. 283–287. Lawrence Erlbaum, Hillsdale (1985)
18. Vega-Pons, S., Correa-Morris, J., Ruiz-Shulcloper, J.: Weighted partition consensus via kernels. *Pattern Recognition* 43(8), 2712–2724 (2010)
19. 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. ICCV*, vol. 2, pp. 416–423 (2001)
20. Rao, S., Mobahi, H., Yang, A.Y., Sastry, S., Ma, Y.: Natural image segmentation with adaptive texture and boundary encoding. In: Zha, H., Taniguchi, R.-i., Maybank, S. (eds.) *ACCV 2009. LNCS*, vol. 5994, pp. 135–146. Springer, Heidelberg (2010)
21. Arbelaez, P., Maire, M., Fowlkes, C.C., Malik, J.: From contours to regions: An empirical evaluation. In: *CVPR*, pp. 2294–2301. IEEE, Los Alamitos (2009)
22. Meilă, M.: Comparing clusterings—an information based distance. *Journal of Multivariate Analysis* 98, 873–895 (2007)
23. Rand, W.M.: Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association* 66, 846–850 (1971)

Real Time Myocardial Strain Analysis of Tagged MR Cines Using Element Space Non-rigid Registration

Bo Li¹, Brett R. Cowan², and Alistair A. Young³

¹ NextWindow Ltd, Auckland, New Zealand

² Auckland Bioengineering Institute, University of Auckland, Auckland, New Zealand

³ Auckland MRI Research Group, University of Auckland, Auckland, New Zealand

Abstract. We develop a real time element-space non-rigid registration technique for cardiac motion tracking, enabling fast and automatic analysis of myocardial strain in tagged magnetic resonance (MR) cines. Non-rigid registration is achieved by minimizing the sum of squared differences for all pixels within a high order finite-element (FE) model customized to the specific geometry of the heart. The objective function and its derivatives are calculated in element space, and converted to image space using the Jacobian of the transformation. This enables an anisotropic distribution of user-defined model parameters, which can be customized to the application, thereby achieving fast estimations which require fewer degrees of freedom for a given level of accuracy than standard isotropic methods. A graphics processing unit (GPU) accelerated Levenberg-Marquardt procedure was implemented in Compute Unified Device Architecture (CUDA) environment to provide a fast, robust optimization procedure. The method was validated in 30 patients with wall motion abnormalities by comparison with ground truth provided by an independent expert observer using a manually-guided analysis procedure. A heart model comprising 32 parameters was capable of processing 36.5 frames per second, with an error in circumferential strain of $-1.97 \pm 1.18\%$. For comparison, a standard isotropic free-form deformation method requiring 324 parameters had greater error ($-3.70 \pm 1.15\%$) and slower frame-rate (4.5 frames/sec). In conclusion, GPU accelerated custom element-space non-rigid image registration enables real time automatic tracking of cardiac motion, and accurate estimation of myocardial strain in tagged MR cines.

1 Introduction

Myocardial strain is an important clinical indicator of regional heart performance. Its main advantage, in comparison with other functional parameters such as ejection fraction, is that it describes the local contraction undergone by the muscle at each point in the heart. Strain and strain rate can be calculated from tissue tagging techniques with magnetic resonance (MR) imaging, such as SPAMM (Spatial Modulation of Magnetization) [1], which allows the noninvasive creation of large number of material tags in soft tissue (Fig. 1).

However, quantitative reconstruction of myocardial deformation and strain from tagged MR data is complex and time-consuming. Filtering methods such as Harmonic Phase (HARP) offers fast processing [12], and Gabor filter banks [11] improved dynamic range using but model-based methods offer the advantages of robust strain estimation [16] and physiologically appropriate regularization [10]. Several model-based analysis techniques have been investigated, which can be classified into active shape methods and non-rigid registration methods. In active shape methods [5,15], features are detected and image forces derived to deform a model to track the features through the temporal sequence [15]. In model-based non-rigid image registration methods [4,14], images are typically deformed to optimize a similarity measure penalizing the difference between a current image and a reference image in order to give an estimate of the underlying deformation [8], which can then be used as an analytical description for strain calculation. A common drawback of these techniques is the intensive computation required to register all images in the sequence, due to the large number of model parameters needed to accurately describe myocardial motion. Li et al. [9] took advantage of commodity graphics processing units (GPUs) to perform the computationally intensive Levenberg-Marquardt optimization procedure in non-rigid registration. However, the main bottleneck is the linear equation solver which is burdened by the large number of parameters in the deformation model (e.g. 324 global control parameters in [9]).

In this paper, we develop a real time non-rigid image registration technique for myocardial strain analysis, based on a user-specified FE model customized to the patient anatomy. A sum of squared pixel intensity differences (SSD) similarity measure is minimized using a GPU-accelerated Levenberg-Marquardt non-linear least squares algorithm. Because the user-specified model can have anisotropic or non-uniform complexity, the parameter distribution can be customised and optimized to the application. Fewer model elements (and model parameters) are therefore required to obtain similar accuracy compared with typical rectangular grid-based isotropic models. Also, robustness is improved as physiologically appropriate regularization constraints can be imposed to achieve physically realistic deformations. Furthermore, the FE model can be used to discard the motion of adjacent structures, or flow artefacts outside the myocardium, thus improving the tracking accuracy.

Significant computational advantages can be achieved if the objective function can be calculated in element space, rather than image space. These include pre-calculation of the FE basis functions, fast determination of the local support domain of each parameter, and efficient scaling with model complexity. The current and reference images are transformed into element space using texture mapping, and the SSD objective function efficiently calculated using GPU algorithms, using the Jacobian of the transformation to map back to standard image space.

This work has similarities to Jordan et al. [6], who also used a FE model combined with non-rigid registration; however, their goal was to provide a modular method for material parameter estimation, whereas ours was to provide a real

time physiologically robust strain estimation method. Chandrashekara et al. [3] describe an efficient subdivision lattice based non-rigid registration procedure which enables a smaller number of control points; however, the objective function was calculated in image space using a time-consuming procedure for locating points within the model, and no attempt was made to provide physiologically meaningful deformation constraints.

Validation of the technique was performed using mid-ventricular short axis images from 30 patients randomly selected from clinical research studies in chronic renal disease and congenital heart disease. Ground truth was provided by an independent expert analyst using a manually guided active shape approach [15]. Errors and frame rates were also compared with a previously validated free-form deformation GPU non-rigid registration method [9].

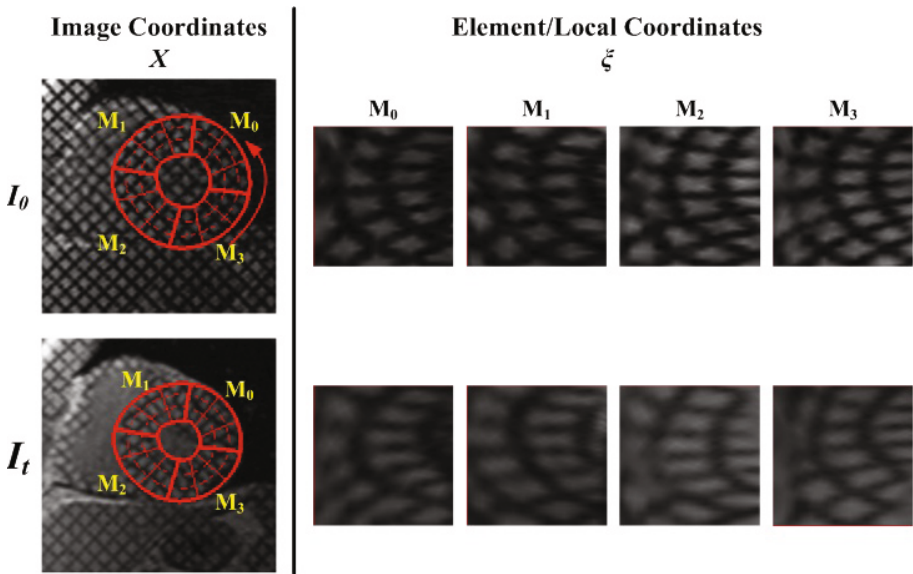


Fig. 1. SPAMM tagged images showing reference (left) and current (right) images in the top panel, and resampled pixels in element coordinate space of the model below. Each model contains four elements: M_0 , M_1 , M_2 , and M_3 . Reference image coordinates: (X, Y) ; current image coordinates: (x, y) ; element coordinates: (ξ_1, ξ_2) .

2 Methods

2.1 Finite Element Model

For short-axis myocardial motion tracking, a FE model was constructed comprising four circumferential elements (Fig. 1), each of which is formed along two local coordinates: a radial axis ξ_1 and a circumferential axis ξ_2 . The initial reference position of the model, M^R , was defined interactively on the first image (frame 0) using guide-point modeling [17]. Briefly, a scale and pose for the model

was defined using the left ventricle (LV) centroid and fiducial markers placed at the insertion of the right ventricle. A small number of epicardial and endocardial boundary points was then fitted by the model, in order to achieve a fast segmentation of the myocardium. The FE description of the displacement field, \mathbf{u} , controlled by a global parameter vector \mathbf{P} , was defined as:

$$\begin{aligned} x(X) &= X + \mathbf{u}(X) \\ \text{where} & \\ \mathbf{u}(X) &= \sum_{\epsilon} \psi^{\epsilon}(\xi(X)) \cdot \mathbf{G} \cdot \mathbf{P} = \sum_{\epsilon} \psi^{\epsilon}(\xi) \cdot P^{\epsilon} \end{aligned} \quad (1)$$

where $X \in M^R$ are reference coordinates, x are the corresponding current coordinates, and $\xi(X)$ maps the reference coordinates X into the corresponding element coordinates ($\xi = [\xi_1, \xi_2]$). The element basis functions $\psi^{\epsilon}(\xi)$ govern the interpolation of the element parameters over the domain of the element. A linear global-to-local parameter map, \mathbf{G} , was used to link the global parameters \mathbf{P} to the local parameters, P^{ϵ} , of each element. Bi-cubic Bézier basis functions with C^1 continuity between the elements were used. Advantages of using C^1 Bézier basis functions [9] include: (1) each global control point has very local support in that only the the four elements around the node are affected, and (2) a multi complexity optimization can be performed using de Casteljau's algorithm for model subdivision.

2.2 Optimization

Given reference and current images I_0 and I_t respectively, the equation for warping of the current image according to \mathbf{u} takes place in reference space with respect to X is:

$$M_t(X) = I_t(x(X)) \quad (2)$$

Thus the warped current image, M_t , is generated by mapping the current image to the reference coordinates.

Different from the work of Li et al. [9] whose similarity measure (or objective function of the registration) was calculated over a rectangular subregion within images, the similarity measure of this work was defined as the SSD of pixel intensities (scaled to $[0, 1]$) over the region of the user-defined model, between the warped current image M_t and reference image I_0 in reference coordinates:

$$E_I = \int_{M^R} (I_0(X) - M_t(X))^2 \cdot dX \quad (3)$$

Complex computations are required in $\xi(X)$ in Equ. (1) to accurately segment pixels in reference coordinates inside or outside the user-defined model. To overcome this problem for computational efficiency, both current and reference images were mapped into element coordinate space of the model according to:

$$\begin{aligned}
 x(\xi) &= X_0(\xi) + \mathbf{u}(\xi) \\
 \text{where} \\
 X_0(\xi) &= \sum_{\epsilon} \psi^{\epsilon}(\xi) \cdot P_0^{\epsilon} \\
 \mathbf{u}(\xi) &= \sum_{\epsilon} \psi^{\epsilon}(\xi) \cdot P_t^{\epsilon}
 \end{aligned}
 \tag{4}$$

where P_0 is the global control parameters of the model at the initial reference position (frame 0), and P_t is the global control parameters of the displacement field from the reference to the current image.

The similarity measure in Equ. (3) can equally derived by the SSD integration over the local element space and multiplied by the Jacobian of the transformation between X and ξ according to the *Change of Variables* theorem:

$$\begin{aligned}
 E_I &= \int_{\xi \in M^R} (I_0(X_0(\xi)) - I_t(x(\xi)))^2 \cdot J(X_0(\xi)) \cdot d\xi \\
 \text{where} \\
 J(\xi) &= \det \begin{pmatrix} \frac{\partial X_0(\xi_1)}{\partial \xi_1} & \frac{\partial X_0(\xi_2)}{\partial \xi_1} \\ \frac{\partial X_0(\xi_1)}{\partial \xi_2} & \frac{\partial X_0(\xi_2)}{\partial \xi_2} \end{pmatrix}
 \end{aligned}
 \tag{5}$$

Thus, all data points corresponding to ξ are implicitly ensured to be within the model domain. Also, this enabled faster calculation of objective function, gradient and Hessian terms since each model parameter has a fixed local control region, determined by pre-calculated FE basis functions $\psi^{\epsilon}(\xi)$, giving rise to a fixed computational cost for these terms independent of the model complexity.

The element coordinates ξ are sampled in the range $[0, 1]$ at a user-specified resolution. In practice the image data and FE model may not be sufficient to regularize the problem, which is ill-posed in the sense of Hadamard. A Sobolev smoothing term [2,7] was included in the error function to provide a physiologically meaningful mechanical constraint on the derived deformation:

$$E_s = \int_{M^R} \left(\alpha_1 \left\| \frac{\partial \mathbf{u}}{\partial \xi_1} \right\|^2 + \alpha_2 \left\| \frac{\partial \mathbf{u}}{\partial \xi_2} \right\|^2 + \beta_1 \left\| \frac{\partial^2 \mathbf{u}}{\partial \xi_1^2} \right\|^2 + \beta_2 \left\| \frac{\partial^2 \mathbf{u}}{\partial \xi_2^2} \right\|^2 + \beta_3 \left\| \frac{\partial^2 \mathbf{u}}{\partial \xi_1 \partial \xi_2} \right\|^2 \right) \tag{6}$$

where $\alpha_1, \alpha_2, \beta_1, \beta_2, \beta_3$ are smoothing weights with values 0.1,1,1,1,10 respectively. These weights were derived according to the expected deformation of the heart. The smallest constraint (α_1) was applied to the stretch along the radial direction since this is typically large and non-homogeneous, followed by stretch and bending in the circumferential direction, with the highest constraint in transverse shear which is typically small. The integration was performed over M^R , the domain of the model.

The objective function $E = E_I + E_S$ was finally optimized using the Levenberg-Marquardt algorithm, as described in [13]. The gradient and Hessian of the

Sobolev smoothing terms were described in [7], whereas the gradient and Hessian of the image term are:

$$\begin{aligned} G_i &= \int_{\xi \in M^R} -2 \cdot I_0(X_0(\xi)) - I_t(x(\xi)) \cdot \frac{I_t(x(\xi))}{\partial P_i} \cdot J(X_0(\xi)) \cdot d\xi \\ H_{ij} &= \int_{\xi \in M^R} 2 \cdot \frac{I_t(x(\xi))}{\partial P_i} \cdot \frac{I_t(x(\xi))}{\partial P_j} \cdot J(X_0(\xi)) \cdot d\xi \end{aligned} \quad (7)$$

where G_i represents the gradient value for parameters i , while H_{ij} represents the Hessian value for parameters i and j . At each iteration, G and H are used to form a system of linear equations, which was then solved using the LU linear equation solver. Note, the Jacobian map, $J(X_0(\xi))$, can be pre-calculated before registration due to the fixed parameter values for X_0 .

2.3 Pattern Detection

In addition, if the underlying pattern of the tracking subject is known in advanced, element-space registration can be modified to automatically locate the reserved pattern/shape within the image. In this case, the similarity measure for pattern detection based on element-space registration becomes:

$$E_I = \int_{\xi \in M^R} (Pat(\xi) - I(x(\xi)))^2 \cdot J(x(\xi)) \cdot d\xi \quad (8)$$

where $Pat(\xi)$ is pre-defined tracking pattern along the local coordinates ξ and I is the image containing the pattern. Therefore, the gradient and Hessian of Equ. (8) are:

$$\begin{aligned} G_i &= \int_{\xi \in M^R} -2 \cdot (Pat(\xi) - I(x(\xi))) \cdot \frac{I(x(\xi))}{\partial P_i} \cdot J(x(\xi)) \cdot d\xi \\ H_{ij} &= \int_{\xi \in M^R} 2 \cdot \frac{I_t(x(\xi))}{\partial P_i} \cdot \frac{I_t(x(\xi))}{\partial P_j} \cdot J(x(\xi)) \cdot d\xi \end{aligned} \quad (9)$$

Since the transformation between local coordinates ξ and image coordinates x is updated at each iteration during registration, the Jacobian map, $J(x(\xi))$, in Equ. (8) needs to be re-calculated at each step of Levenberg-Marquardt optimization.

This pattern detection technique has been validated in a chessboard corners detection application, which is widely used for calibrating the intrinsic and extrinsic parameters of video cameras [18].

As shown in (Fig. 2), a flashing infrared light source was synchronized with image acquisition, (a) and (b), thus pixels on the chessboard were successfully isolated from the whole image. Then, a 2×2 FE model was initially positioned over the region of chessboard pixels, which was determined by simply looking for the most left, right, bottom and top pixels over the entire image, (c). Further, element-space registration minimize the SSD between pixels values along model's local coordinates, (f), and the desired chessboard pattern, (e), and finally fitted the FE model to the region of chessboard. (g) in Fig. 2 represents the pixel intensity along the local coordinates of the fitted model.

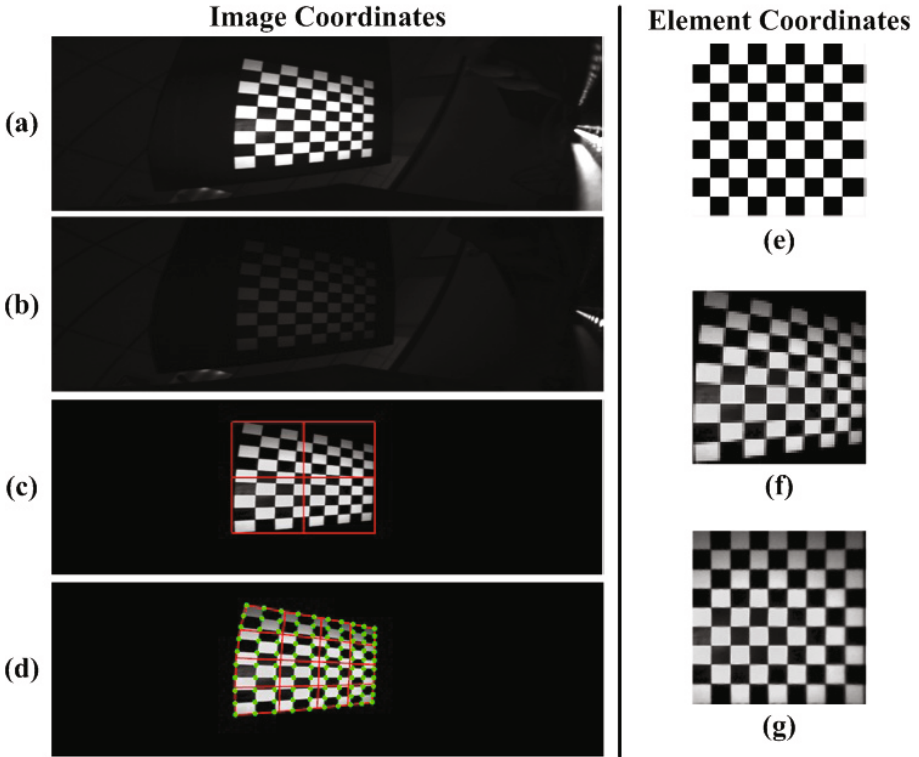


Fig. 2. Chessboard corners detection using element-space registration. (a) frame with light on; (b) frame with light off; (c) background removed image with initial position of a 2×2 FE model (red); (d) model fitted to the image with detected corners (green); (e) reference chessboard pattern in the local coordinates; (f) pixels at the local coordinates of the initial FE model; (g) pixels at the local coordinates of the fitted FE model.

2.4 Cardiac Motion Tracking

For tagged MRI images, since the pattern over the myocardial region of the heart is not as fixed as the chessboard pattern, our technique for cardiac motion tracking still require user's input at the first frame of the cine to segment the myocardial region, then we can use the pattern within the segmented region to calculated the displacement of myocardium from the first frame to the next frame, and so forth.

In comparison to the incremental motion tracking technique in [9], since element-space registration can begin from any shape of the user-specified model, its initial model geometry, X_0 during registration at each consecutive frames, is progressively updated by the geometry from the registration in previous frames. Therefore, it provides a better estimation of myocardium region for images at current frames than a fixed initial model geometry. For example, starting from the user-defined model in frame 0 (M_0), after registering the image at frame

1 to 0 ($M_1 = M_0 + dM_0$), we use the resultant geometry M_1 as the starting solution to register from frame 2 to 1 (dM_1). Since the region of model in the registration of the next consecutive frames is set to the previous solution, the model maintains its correspondence with the object. Continuously applying this registration process to all frames directly lead to the warped positions at each frame, so there is no need for an additional process to accumulate displacement fields. Also, all material points at the first frame, $X \in M_0$, are guaranteed to find a one-to-one mapping in the rest of frames due to the same local coordinates ξ .

The problem of accumulating tracking errors in [9] also persisted in the element-space motion tracking: assume X_0 has local coordinates (ξ_1, ξ_2) at frame 0, and its actual corresponding point is x_1 at frame 1. If an error occurs that derives $x'_1 = x_1 + error$ at frame 1, the motion tracking process will register (ξ_1, ξ_2) to X'_1 , and this *error* remains in the following registration throughout the entire cine. Although the bi-directional motion tracking ([9]) was reported as a robust method to ameliorate accumulated tracking errors from each direction of the cine, we still insist on forward registration due to the significant feature changes between first and last frames of tagged MR image sequence caused by T1 tag fading.

2.5 CUDA Implementation

The GPU version of the proposed method was implemented in the Compute Unified Device Architecture (CUDA) environment using an Nvidia Geforce 8800 GTS graphics card, with 16 multiprocessors, each containing 8 stream processors, for a total of 128 processors to perform data-level parallelism. The schema of the CUDA implementation was similar to the Cg implementation described in [9]. All tasks in the registration procedures were divided into CPU and GPU components, in which the CPU components mainly involved: (1) off-line and once only pre-calculations; (2) linear equation solution using conjugate gradients; and (3) check of convergence based on the similarity measure. The GPU components included three CUDA subprograms, which performed the similarity measure, gradient calculation, and Hessian calculation respectively. Both CPU and GPU components formed a loop in the schema to facilitate the iterative optimization procedure.

In contrast to the GPU implementation in [9], our registration requires the similarity measure to be calculated in the element space of the model, rather than the image space. Therefore, the global coordinates at each thread in the CUDA subprogram were firstly converted into element coordinate values (scaled to $[0, 1]$ in each direction) at a user-specified resolution, secondly passed to the FE model to derive its position in the image coordinates, and finally mapped to pixel values in current image coordinates by a CUDA texture lookup function. Furthermore, our registration required pre-calculation of the Jacobian map from the reference model, and inclusion of Jacobian factors in the similarity measure, gradient, and Hessian calculations.

2.6 Experiments

The method was applied to mid-ventricular short axis images in 30 patients randomly selected from clinical research studies in chronic renal disease and congenital heart disease. Ground truth was provided by an independent expert analyst using a manually guided active contour approach [15]. Briefly, a 2D FE model with four circumferential elements was manually customized to the inner and outer boundaries of the left ventricle at the end-diastolic (ED) frame and using guide-point modeling [17]. The model was then deformed to track each stripe from frame 0 to the rest of frames using active contours with manual correction [15]. In order to validate the automated tracking algorithm, the manually defined FE model at ED was used as the starting model for the FE non-rigid registration throughout the rest of cardiac cycle. The average circumferential strain error at the end-systolic frame (the frame with maximal error over all frames) was then calculated between manual and automatic results.

3 Results

The mean circumferential strain by the expert analyst at end-systole was $-19.1 \pm 3.0\%$. Fig. 3 presents the agreement between the results from the automated method and ground truth over 30 patients with a Bland-Altman plot, in which the x-axis is the average circumferential strain between the two measurements and the y-axis is the difference between them. One case had a large error (-6.0%) error due to incorrect placement of the initial contours at frame 0. The average error over the entire dataset is $(-1.97 \pm 1.18\%)$. For comparison, a standard free-form deformation image registration method [17] with 64 elements (324 model parameters) resulted in larger inconsistency ($-3.70 \pm 1.15\%$) from ground truth.

The speed of the CUDA implementation was compared against a fully CPU implementation and the GPU accelerated standard isotropic model described in [9]. The CPU element-space registration (32 parameters) ran at 4.6 frames/sec,

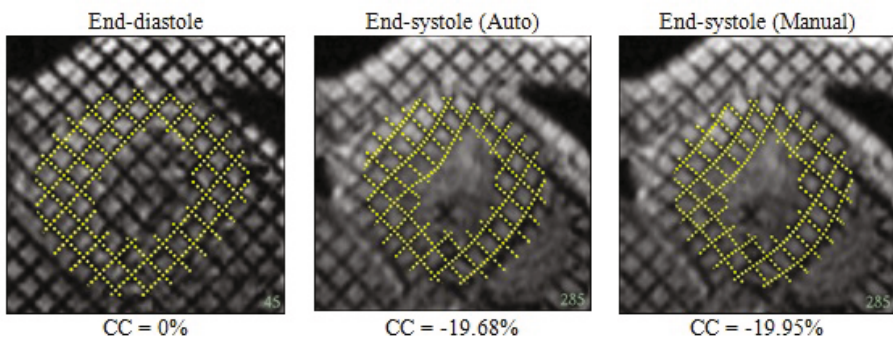


Fig. 3. Automatic and manual tag tracking result at the end-systole (time of smallest blood volume) frame using element coordinates registration based on the user-specified model at the end-diastolic frame (frame 0), in a patient with a systemic right ventricle

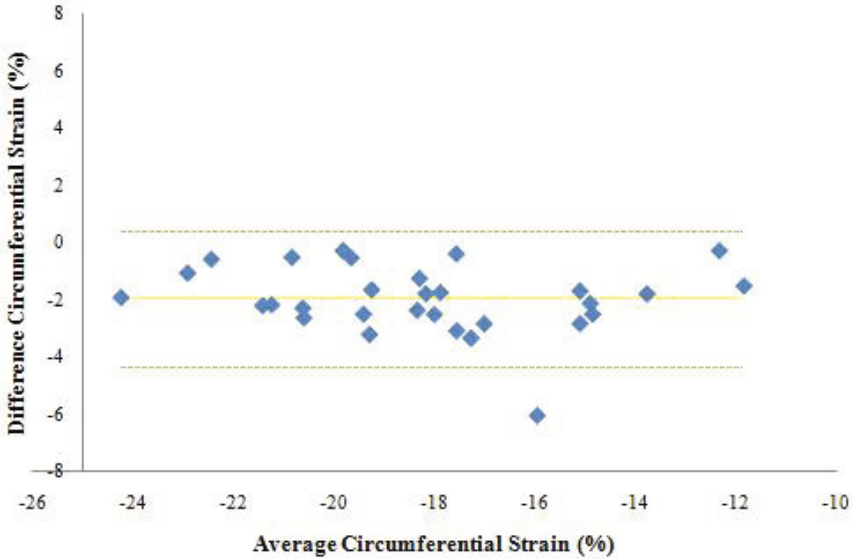


Fig. 4. The Bland-Altman plot of the calculated circumferential strain at end-systolic frame using finite element non-rigid registration against ground truth. Lines show the range of average \pm two standard divisions of the data.

whereas the GPU accelerated isotropic model with 324 parameters ran at 4.5 frames/sec. The CUDA element-space registration procedure (32 parameters) was about 8 times faster at 36.5 frames/sec.

4 Discussion and Future Work

In this paper, a element-space non-rigid registration method was developed for real time automated tag tracking and strain analysis. It gave similar values for the circumferential strain at the end-systolic frame as a previously validated expert analysis. In comparison to other non-rigid image registration techniques [14,4,9], the approach uses an anatomically customized model for image warping, which has better distribution of model parameters (anisotropic) enabling better solutions with fewer parameters. Since our approach significantly reduces the complexity of the transformation model, it provides a faster solution both in theory and practice. Furthermore, the calculations of the similarity measure, gradient and Hessian in element coordinates enabled better use of advanced graphics hardware to perform non-rigid registration calculations. The limitations of our method, as with many other similar methods, are that its performance depends mainly on the quality and the signal to noise ratio of the images. Also, in motion tracking, the tracking error existing between two consecutive frames is passed to the following frames and this can be hard to correct. The bias of -2% may be due to over-smoothing, since the weights were not extensively

optimized in this work. However, the variability of 1.2% is excellent compared with published inter-observer errors [16].

In future work, we are planning to extend the FE non-rigid image registration to three dimensions and to the time domain, by integrating time domain parameters into the deformable model. Also, more effort will be spent on investigating an uniform pattern at the myocardium for pattern detection using element-space registration.

Acknowledgements. This work was supported by the Health Research Council of New Zealand. We are grateful to Dr Nicky Edwards and Dr Jonathan N Townend of the University of Birmingham for the use of the images.

References

1. Axel, L., Dougherty, L.: MR imaging of motion with spatial modulation of magnetization. *Radiology* 171, 841–845 (1989)
2. Beg, M.F., Miller, M.I., Trounev, A., Younes, L.: Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International Journal of Computer Vision* 61, 139–157 (2005)
3. Chandrashekhara, R., Mohiaddin, R., Razavi, R., Rueckert, D.: Nonrigid image registration with subdivision lattices: Application to cardiac mr image analysis. In: Ayache, N., Ourselin, S., Maeder, A. (eds.) *MICCAI 2007, Part I. LNCS*, vol. 4791, pp. 335–342. Springer, Heidelberg (2007)
4. Chandrashekhara, R., Mohiaddin, R.H., Rueckert, D.: Analysis of 3-D myocardial motion in tagged MR images using nonrigid image registration. *IEEE Transactions on Medical Imaging* 23, 1245–1250 (2004)
5. Declerck, J., Feldmar, J., Ayache, N.: Definition of a four-dimensional continuous planispheric transformation for the tracking and the analysis of left-ventricle motion. *Medical Image Analysis* 2, 197–213 (1998)
6. Jordan, P., Socrate, S., Zickler, T.E., Howe, R.D.: A Nonrigid Image Registration Framework for Identification of Tissue Mechanical Parameters. In: Metaxas, D., Axel, L., Fichtinger, G., Székely, G. (eds.) *MICCAI 2008, Part II. LNCS*, vol. 5242, pp. 930–938. Springer, Heidelberg (2008)
7. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. *International Journal of computer vision* 1, 321–331 (1988)
8. Ledesma-Carbayo, M.J., Derbyshire, J.A., Sampath, S., Santos, A., Desco, M., McVeigh, E.R.: Unsupervised estimation of myocardial displacement from tagged MR sequences using nonrigid registration. *Magnetic Resonance in Medicine* 59, 181–189 (2008)
9. Li, B., Young, A., Cowan, B.: GPU accelerated non-rigid registration for the evaluation of cardiac function. In: Metaxas, D., Axel, L., Fichtinger, G., Székely, G. (eds.) *MICCAI 2008, Part II. LNCS*, vol. 5242, pp. 880–887. Springer, Heidelberg (2008)
10. McInerney, T., Terzopoulos, D.: Deformable models in medical image analysis. In: *Proceedings of the Workshop on Mathematical Methods in Biomedical Image Analysis*, pp. 171–180 (1996)
11. Montillo, A., Metaxas, D., Axel, L.: Extracting tissue deformation using gabor filter banks. In: *Proc. of SPIE*, vol. 5369, p. 2 (1996)

12. Osman, N.F., Kerwin, W.S., McVeigh, E.R., Prince, J.L.: Cardiac motion tracking using CINE harmonic phase (HARP) magnetic resonance imaging. In: Proceedings of the Workshop on Mathematical Methods in Biomedical Image Analysis, vol. 42, pp. 1048–1060 (1999)
13. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical recipes: the art of scientific computing. Cambridge University Press, Cambridge (2007)
14. Wierzbicki, M., Drangova, M., Guiraudon, G., Peters, T.: Validation of dynamic heart models obtained using non-linear registration for virtual reality training, planning, and guidance of minimally invasive cardiac surgeries. *Medical Image Analysis* 8, 387–401 (2004)
15. Young, A.A.: Model tags: Direct 3D tracking of heart wall motion from tagged MR images. In: Wells, W.M., Colchester, A.C.F., Delp, S.L. (eds.) MICCAI 1998. LNCS, vol. 1496, pp. 92–106. Springer, Heidelberg (1998)
16. Young, A.A., Axel, L., Dougherty, L., Bogen, D.K., Parenteau, C.S.: Validation of tagging with MR imaging to estimate material deformation. *Radiology* 188, 101–108 (1993)
17. Young, A.A., Cowan, B.R., Thrupp, S.F., Hedley, W.J., DellItalia, L.J.: Left Ventricular Mass and Volume: Fast Calculation with Guide-Point Modeling on MR Images. *Radiology* 216, 597–602 (2000)
18. Zhang, Z.: A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence* 22, 1330–1334 (2000)

Extending AMCW Lidar Depth-of-Field Using a Coded Aperture

John P. Godbaz, Michael J. Cree, and Adrian A. Dorrington

School of Engineering, University of Waikato, Hamilton, New Zealand
jpg7@waikato.ac.nz

Abstract. By augmenting a high resolution full-field Amplitude Modulated Continuous Wave lidar system with a coded aperture, we show that depth-of-field can be extended using explicit, albeit blurred, range data to determine PSF scale. Because complex domain range-images contain explicit range information, the aperture design is unconstrained by the necessity for range determination by depth-from-defocus. The coded aperture design is shown to improve restoration quality over a circular aperture. A proof-of-concept algorithm using dynamic PSF determination and spatially variant Landweber iterations is developed and using an empirically sampled point spread function is shown to work in cases without serious multipath interference or high phase complexity.

1 Introduction

Full-field amplitude modulated continuous wave (AMCW) lidar systems utilise the time-of-flight (TOF) principle to generate two dimensional matrices of intensity and radial range values using active scene illumination. Whereas point and line scanner based systems require expensive mechanical systems to sequentially capture a point cloud, full-field systems capture an entire image simultaneously and near-instantly opening up a variety of applications including games, medical imaging, security and engineering quality control.

However, despite their advantages, full-field AMCW systems introduce new challenges such as systematic errors due to multipath interference and limited depth-of-field (DOF). In full-field AMCW lidar systems limited DOF results in both erroneous range and intensity values around the edges of objects as well as a loss of detail. While most previous computational photography work has addressed the DOF problem for intensity images using techniques such as coded apertures [1] and plenoptic cameras [2], previous systems have relied on implicit range information. Since AMCW lidar systems produce explicit range information, albeit limited by the DOF, there is inherently more information available to assist in restoration.

Prior depth-from-defocus (DFD) techniques [3,4] utilise the known range variant properties of the PSF to determine distance, however typically require more than one image of a scene. More modern methods have used coded apertures to make the blurring less of a low-pass filter and enable high quality restoration while requiring only a single image [1]. Related work has changed the nature

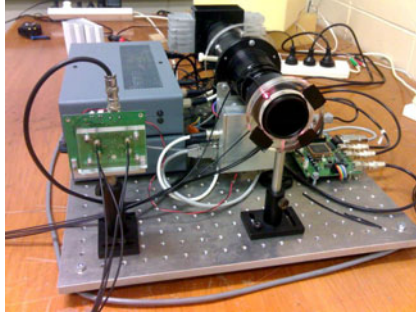


Fig. 1. Our full-field lidar system. While it may initially appear bulky and inelegant, it provides capabilities that existing commercial systems do not. In the configuration shown three of the four illumination sources are occluded. For this paper, all four illumination sources were utilised to provide coaxial illumination of the scenes.

of motion blur using coded fluttered shutter patterns [5]. Traditional plenoptic cameras allow refocussing without any explicit range information [2] however sacrifice spatial resolution. Alternative methods like Lumsdaine and Georgiev’s ‘Plenoptic 2.0’ [6], which offer a substantial increase in resolution, require the determination of a range dependent magnification parameter in order to produce an artefact free image. Other techniques for defocus invariance include wavefront coding [7] and merging multiple images at different focal settings. Deconvolution techniques have been previously applied to full-field lidar images for the purposes of light scattering reduction [8,9]. Another work [10] blindly determined the focal parameters of a full-field lidar system and utilised them to improve DOF.

In this paper we briefly demonstrate the advantages of our coded-aperture design over a circular aperture for extending DOF and then show the deconvolution of real defocussed range-images captured using a coded-aperture variation of the full-field heterodyne AMCW lidar system from [11]. A picture of the system is given in fig. 1.

2 Background Theory and System Design

2.1 AMCW Lidar

AMCW lidar systems work by illuminating a target scene with modulated light and then sampling the correlation of the reflected light with a reference signal at the same or a slightly different frequency. The TOF results in a range variant phase shift in the returned illumination – this phase shift is typically measured by mixing the returned light with a reference signal using either a modulated CCD or CMOS sensor [12] or modulated image intensifier [11].

An image intensifier is typically used in devices like night vision goggles to amplify light intensity across a 2D field of view. By modulating the image intensifier gain at high frequency, it is possible to optically correlate the reference signal

with the backscattered illumination modulation signal. A technique known as heterodyning allows the difficult, high frequency phase measurement problem to be reduced to an easier low frequency phase measurement problem. If the illumination modulation signal is at x Hz and the reference modulation is at y Hz, then a downconverted correlation waveform is formed at $(x - y)$ Hz. Since the phase offset of the downconverted correlation waveform is proportional to that of the backscattered illumination signal, if $(x - y)$ Hz is sufficiently low then phase can be calculated from data captured using an off-the-shelf CCD camera.

2.2 The Range-Imager

Fig. 2 shows the optical configuration of the ranger system. The scene is illuminated by modulated laser light and imaged by a Nikkor 50mm f/1.8D lens where the aperture diaphragm blades are replaced with a coded aperture. The primary optics image the scene onto the mirror-like surface of the image intensifier photocathode. A phosphor screen displays the correlation of the returned scene illumination with the image intensifier modulation signal. This results in a temporally varying correlation waveform, where phase corresponds to object range. The phosphor screen is focussed onto a CCD image sensor using additional coupling optics, thus allowing the measurement of range and active intensity.

Raw range information is typically encoded as complex domain values and is generated by calculating the bin of the temporal discrete Fourier transform corresponding to the correlation waveform fundamental frequency for each pixel. This value corresponds to a sample of a particular bin of the spatial Fourier transform of component signal returns over range. For a single pixel composed of a single component return an ideal AMCW lidar measurement can be written as

$$\eta = ae^{4\pi jd/\lambda} \quad (1)$$

where $\eta \in \mathbb{C}$ is a complex domain range measurement, a is the active intensity, d is the distance from the camera and λ is the illumination modulation wavelength.

In practice, AMCW lidar measurements are subject to systematic errors, particularly due to the impact of multipath interference. Multipath interference, of which mixed pixels are a type, is caused when a single pixel integrates light from sources at more than one range causing an erroneous range measurement – the erroneous value being the sum of the complex domain range measurements of each component return. This can result in range-intensity coupling, where the measured range is a function of intensity. When a range image is subject to limited DOF, blurring of the edges of objects results in the formation of large bands of mixed pixels containing erroneous values. One of the aims of this paper is to demonstrate that these erroneous values can be restored. Methods have been developed to mitigate [13] or find the component returns within mixed pixels [14, 15], however the output from these algorithms is difficult to incorporate into a simple deconvolution model. Since each component within a mixed pixel is at a different range from the camera, each has a different PSF. For this paper we model each pixel as being at a single discrete range, which while non-ideal, retains the simplicity of a single two dimensional image array.

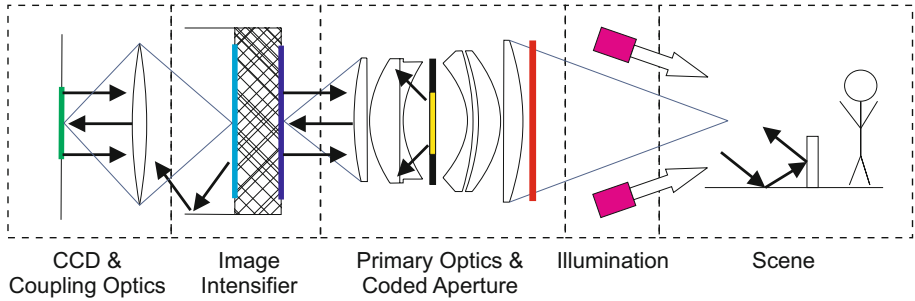


Fig. 2. The optical configuration of the range-imager. Key: modulated lasers (magenta), narrowband filter (red), coded aperture (yellow), image intensifier photocathode (blue), phosphor screen (cyan), CCD image sensor (green). Black arrows represent sources of multipath.

At the moment full-field lidar image processing research is limited by the unavailability of off-the-shelf high resolution systems and the black-box nature of many commercial devices. The custom range-imager utilised for this paper has an effective resolution of around 200,000 pixels – many times that of any commercially available device. However, this comes at the cost of an increase in complexity due to the additional optics required to couple the image intensifier to the CCD and an increase in scattered light.

2.3 Image Formation

From geometric optics, the defocus PSF for an optical system is a scaled image of the aperture shape given by

$$r_p = \alpha \left(1 - \frac{\beta}{d} \right) \quad (2)$$

where r_p is the radius of the PSF, d is the distance from the first principal plane to the object, β is the distance from the first principal plane to the point on the optical axis at which objects are most in-focus and α is a scaling constant [10]. In the Fourier domain, convolution by a PSF corresponds to elementwise multiplication of the spatial frequencies of the image with the spatial frequencies of the PSF

$$g = f \star h \Leftrightarrow G[u, v] = F[u, v]H[u, v] \quad (3)$$

where f is the original image, g is the blurred image and h is the PSF. Any spatial frequencies missing from the PSF are lost, making high quality image restoration difficult. A standard pillbox PSF is non-ideal because it has zeros in its MTF. A coded aperture works by inserting a device into the light path that changes the effective aperture, generally with the aim of improving the properties of the MTF. By whitening the MTF it is possible to improve the quality of restored images. Because there is explicit range information, it is possible to aim for as

broadband a PSF as possible without the constraints imposed by extraction of implicit range information.

3 Methodology

3.1 The Coded Aperture

The coded aperture utilised for this paper is a 7×7 random noise pattern that was printed onto an overhead projector transparency (OHT) as shown in fig. 3. Due to the limited contrast provided by the printing process, the aperture pattern was augmented using marker pen – this resulted in slight unevenness, but had no other impact due to empirical sampling. Advantages of this method of aperture construction include low cost and that any pattern can be produced without physical constraints such as the connectivity required for a physical cut-out pattern. The biggest disadvantage is that depending on the type and quality of the OHT material, the aperture may contribute to light scattering and reflection within the ranger. Some previous approaches include cut out patterns [1] and LCD screens [16].

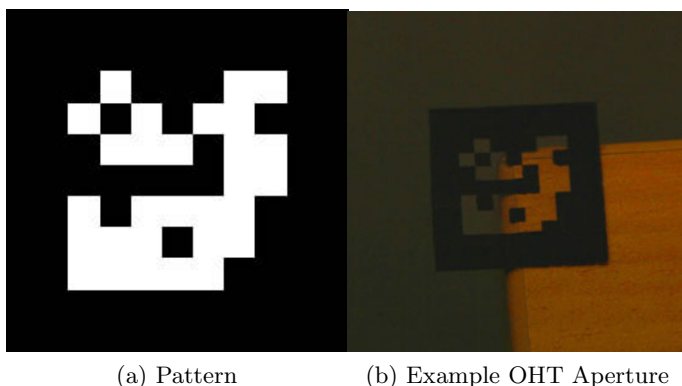


Fig. 3. The binary pattern utilised for this paper and an example OHT based coded aperture

In order to compare our coded aperture design to a similarly sized circular aperture we simulated blurred and noisy intensity and phase images. Fig. 4 shows how the coded aperture improves the performance of deconvolution for an intensity image. For the Lena image at a SNR of 1000 : 1 there is a 24% decrease in RMS error in the restored image. Fig. 5 shows how the coded aperture affects the restoration of phase content in a pure phase image – that is a simulated range image where every pixel has a modulus of one, thus isolating the impact on phase information. The blurred phase information for the textured object counterintuitively appears to peak where there are troughs in the unblurred image due to the black regions in the centre of the aperture pattern. Despite designing the aperture for a white spectral response, limited Gibbs’ phenomenon still occurs at hard discontinuities.

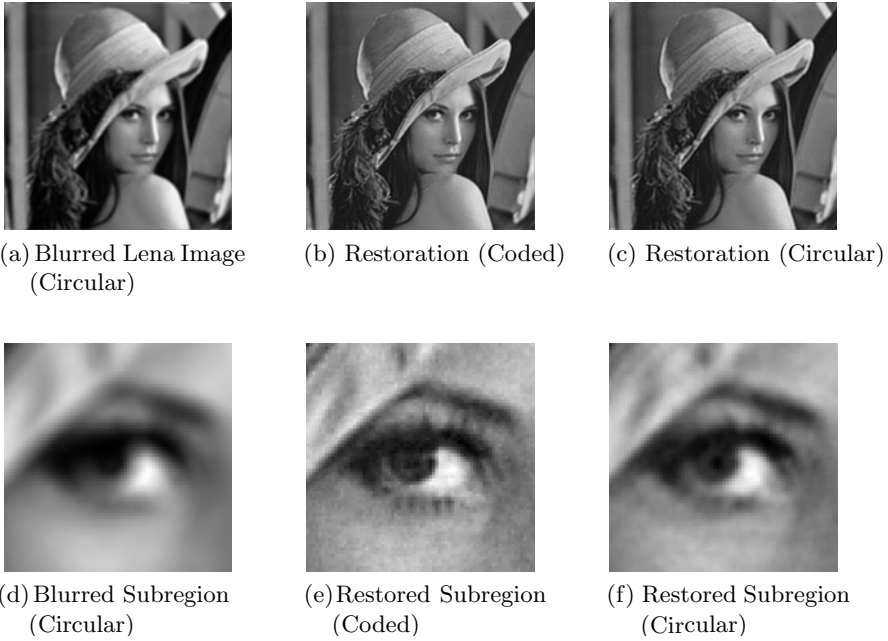


Fig. 4. The impact of aperture choice on deconvolution restoration quality of an intensity image in the known, isoplanatic PSF case. Simulated at a SNR of 1000 : 1, $\lambda = 0.015$ with 50 Landweber iterations.

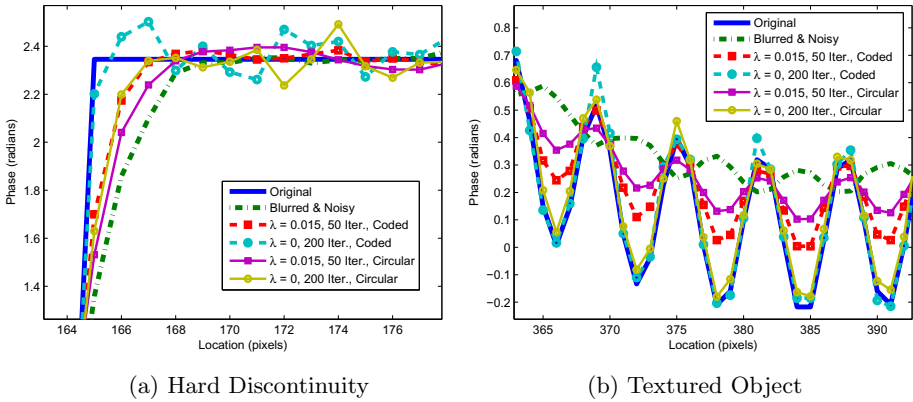


Fig. 5. Slices through a simulated pure phase image pre- and post-deconvolution using a SNR of 1000 : 1. For a given regularisation constant the coded aperture generally results in better restoration quality than a circular aperture – the behaviour for the phase of a complex number is similar to that in the case of an intensity image, but with a slightly greater sensitivity to ringing.

3.2 The Point Spread Function

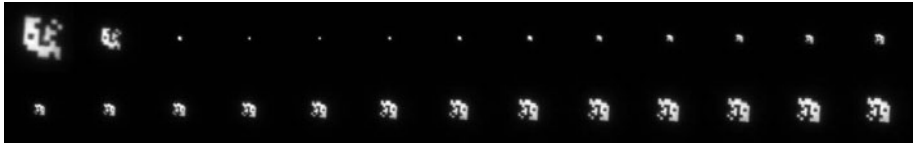
The empirical point spread function of our system is formed as the convolution of the fixed point spread function of the image intensifier and CCD coupling optics with the range-variant point spread function of the primary optics. The image formation process for an AMCW range-imager is the same as for a standard camera with the exception that any reflections before the image intensifier result in an increased TOF and thus a phase shift in the range measurements; fully modelling this requires the utilisation of a complex domain PSF.

Previous papers have sampled the PSF of a full-field AMCW lidar system – both for the purpose of extending DOF [10] and for the purpose of mitigating multipath due to scattering in the range-imager optics [8,9]. While [9] utilised retroreflective dots, we utilise a fibre-optic based point source because it offers better performance while remaining subpixel in size. Attempting to measure both the defocus PSF and scattering effects at the same time is very difficult due to the extreme dynamic range required. In particular, temperature stability is extraordinarily important because even a slight change in bias can result in a massive redistribution of intensity from the defocus component of the PSF to the scattering component.

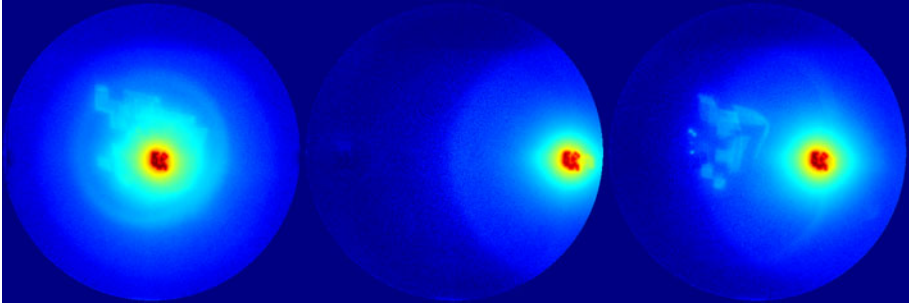
Fig. 6a shows how the PSF changes over range. Allowing for the image intensifier and coupling optics, the PSF scales in the manner predicted by eqn. 2. However the PSF samples close to the ranger are much more blurred than the PSFs of similar radius at a large distance – possible causes include optical aberrations and scattering from the coded aperture. There is a slight pincushion effect on the PSF shape due to radial distortion from the component lenses.

The PSF also changes spatially; fig. 6b shows the log intensity of the PSF in order to highlight subtle scattering effects. Most notably, there is an inverted image of the coded aperture present in the left-most image, which distorts and disappears as the point source is moved to the right side of the image – there is also a soft halo and some specular ‘dots’ (right-most image). Because of the spatial complexity of the PSF, we only utilise centred PSF samples, otherwise the large number of PSF samples would greatly increase the computational complexity of the restoration.

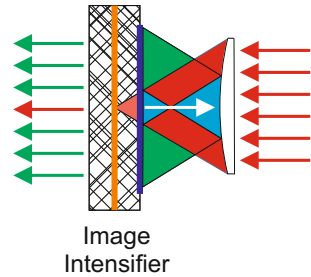
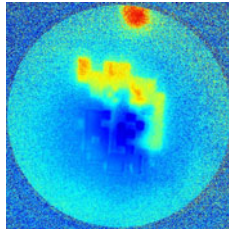
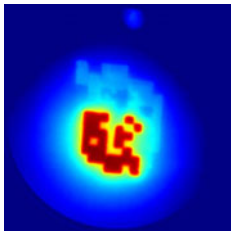
Calculating the phase of extremely dark scattered light is very difficult, so barring inordinately long exposure times or image intensifier burn-in due to oversaturation it is only possible to image the complex domain PSF with extreme defocus. High levels of defocus allow the intensity of the scattered light to be increased while keeping the maximum image intensity to a safe region for the image intensifier. Thus while we still model scattering, we cannot plausibly model the slight phase shifts inherent in the scattering PSF across the entire PSF gamut. Fig. 6c shows the complex domain PSF for an extremely defocused point source – note the low SNR for the darkest regions. Since the point source is within a few centimetres of the optics, the path length difference for light travelling through different sections of the aperture is visible – the path length varies by almost a centimetre within the primary/defocus PSF (blue/cyan). The reflections in the background have a much greater path length; the inverted



(a) PSF Range Variance (Intensity)



(b) PSF Spatial Variance (Log Intensity)



(c) Complex PSF – Log Intensity (left), Phase (right) (d) PSF Formation Model

Fig. 6. Spatial and range variation in the coded-aperture range-imager PSF. In addition, the complex domain PSF is shown for the highly defocussed case – showing subtle phase shifts in the scattered light. In log-intensity images, red represents high intensity and blue low. In phase images, red represents greater distance and blue less distance. From these data we can determine the formation process for the most prominent scattering. In fig. 6d, the initial aperture image (red) is reflected off the image intensifier and back to the final lens in the primary optics (cyan). Despite the low reflectivity of the lens, a significant amount of light is reflected back towards the image intensifier (green). The focal plane (orange) moves as the range to the point source changes, thus changing whether the primary PSF is inverted and the size of both the primary and reflected PSFs. The reflected PSF always has the same orientation.

aperture shape (yellow) has a path length at least 6cm longer than the primary PSF and the reflection at the top (red) has a path length at least 7.5cm longer. From this information, we can determine the formation process for the inverted image – this is given in fig. 6d. We are unaware of any previous measurements of the complex domain PSF of a full-field AMCW lidar system.

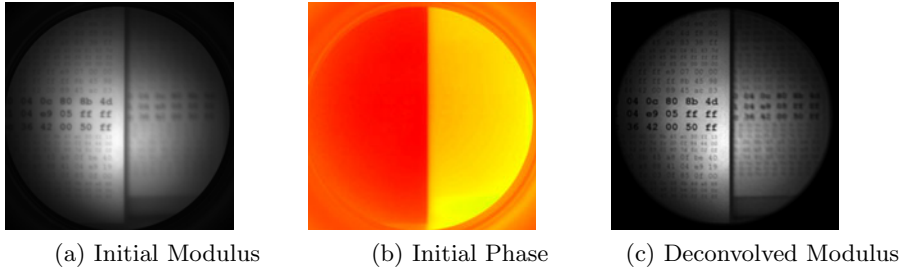


Fig. 7. Scene One, pre- and post-deconvolution. For the phase image, red represents objects closer to the camera (smaller phase offset) and yellow objects farther away (greater phase offset). The restoration of the hard phase discontinuity is shown in fig. 9a.

3.3 Restoration Method

We use a spatially variant Landweber [17] deconvolution method using a Gaussian spatial derivative prior and a weighting mask to remove boundary effects due to the image intensifier. By writing the spatially variant convolution as a matrix transformation, $f \star_{sv} h = Tf$, each iteration becomes

$$\hat{f}_{n+1} = \hat{f}_n + \gamma(T^*W(g - T\hat{f}_n) - \lambda L\hat{f}_n) \tag{4}$$

where \hat{f}_n is the estimate of the unblurred image at the n th iteration, $*$ is the Hermitian transpose of a matrix, γ is a gain term, W is a diagonal matrix of data weights, λ is the regularisation parameter and L is a Laplacian kernel. This is equivalent to iteratively minimising the function

$$\epsilon(\hat{f}) = \|W(g - T\hat{f})\|_2^2 + \lambda\|D_h\hat{f}\|_2^2 + \lambda\|D_v\hat{f}\|_2^2 \tag{5}$$

using gradient descent, where D_h is a horizontal derivative filter and D_v is a vertical derivative filter. The initial estimate is the captured blurred range-image. Additional blank, zero weighted boundaries are added to each image, increasing the image size from 512×512 to 768×768 to mitigate wraparound effects from the use of circular convolutions.

Before each iteration the PSF is dynamically determined for each pixel using radial distance calculated from the phase angle of value in \hat{f}_n . In general, distance along the optical axis can be approximated without calibration by the radial distance. A threshold is set for each restoration, usually 10 iterations, at which point the PSF stops being dynamically updated to prevent noise amplification. This method typically works quite well in regions with edge induced mixed pixels as the values tend to converge to a sharper edge, but in regions subject to severe range-intensity coupling due to scattered light the algorithm can fail.

4 Results and Discussion

Three different scenes were imaged of increasing spatial complexity: two boxes at varying distance from the ranger (fig. 7), a garden gnome and several patterned

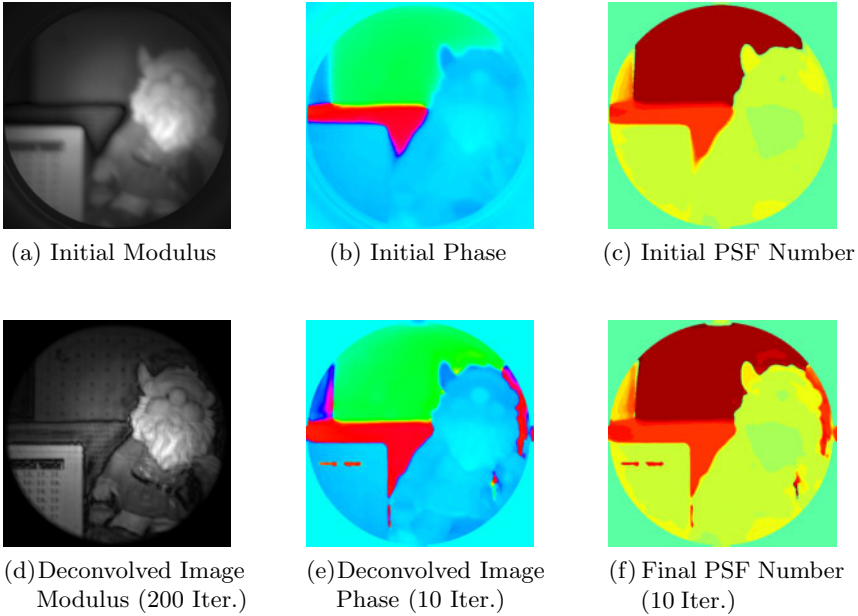


Fig. 8. Scene two, pre- and post-deconvolution. For the phase images, hue represents phase and is cyclic – in order of increasing phase: cyan, blue, magenta, red, yellow, green, cyan. Due to the high modulation frequency, the depth of the scene exceeds the ambiguity interval. While a large number of iterations increases modulus resolution substantially, it tends to introduce unnecessary ringing into phase information.

boards (fig. 8) and a chess set (fig. 10). Due to the optical configuration, ground truth was unavailable. Slices through the first two scenes are shown in fig. 9.

Scene one is an extremely simple scene containing two boxes printed with a test pattern. Fig. 7a shows the initial blurred modulus, which using the blurred range information from fig. 7b is restored to the point where most of the text can be read – a substantial improvement in DOF. Fig. 9a shows how the phase is recovered during the deconvolution process – this graph shows a horizontal slice through the scene in the middle. The deconvolution process results in a substantial sharpening of the boundary between the two boxes as well as a significant shift in the range of the right hand box due to the partial removal of some scattered light. However there remains range-intensity coupling post-deconvolution most probably due to incomplete modelling of the spatial variance of scattering. It is extremely common in real images for range measurements to be shifted by 2-3cm due to scattered light.

Scene two is a more complicated scene. Due to the larger dynamic range, the modulus images of both scenes two and three use gamma compression of $\gamma = 0.5$. In this scene there is much more significant blurring and light scattering. Fig. 8c shows the initial PSF used for each pixel, by the 10th iteration the PSF has changed in regions such as between the garden gnome and front-most board (fig. 8f). In the final deconvolved range-image the modulus (fig. 8d) and phase

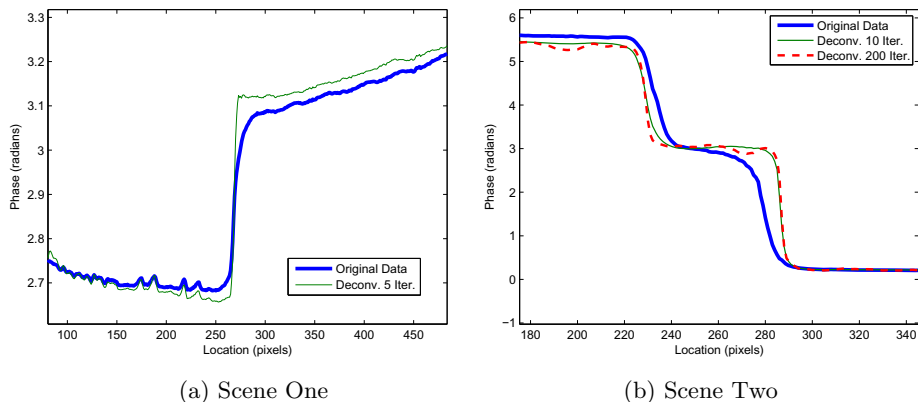


Fig. 9. The phase of slices through scenes 1 and 2, before and after deconvolution. From a phase perspective, 200 iterations provides few benefits over 10 iterations. Fig. 9a shows range-intensity coupling before and after restoration.

(fig. 8e) components have substantially improved sharpness, although there are some notable artefacts. Most noticeable is the erroneous range value given for the black tape holding the test pattern onto the front board – the red range value is roughly equivalent to phase shifting the correct range value by π radians and this may indicate excessive compensation for scattering. There are ringing effects around the edges of objects such as the head of the gnome and the pattern. Like many real-life range-images, scene two contains a small region at the top left which is outside the range ambiguity interval – ie. due to the modulo 2π nature of phase, this region has been deconvolved by an incorrect PSF. This is unavoidable for real-world scenes unless range precision is sacrificed by using a particularly low modulation frequency or a phase unwrapping method utilised.

Unlike normal intensity images, complex domain range-images have some complicated behaviour around edges. In typical scenes the edges of objects are mixed pixels, however these tend to be heavily attenuated by the deconvolution process, resulting in dark bands at the boundaries of objects. A different type of dark band is seen in defocussed images where the objects have sufficiently different phases as to result in partial cancellation – these bands can be seen around the edges of the chess pieces in fig. 10a. While a smoothness constraint may limit the impact of noise on the restoration, it also has a tendency to intensify dark bands between objects at significantly different ranges. If the aim of a restoration is to produce an in-focus pure intensity type image, then it may be more appropriate to deconvolve the total integrated intensity, which is essentially the total amount of light detected by the ranger. Albeit, most commercial ranger-imagers use a differential measurement process that removes this information from the raw measurements.

Scene three demonstrates the current limitations of the restoration algorithm. The extreme range-intensity coupling is demonstrated by the black chess pieces. Regions such as the knight’s head, which is near black, are perturbed by light

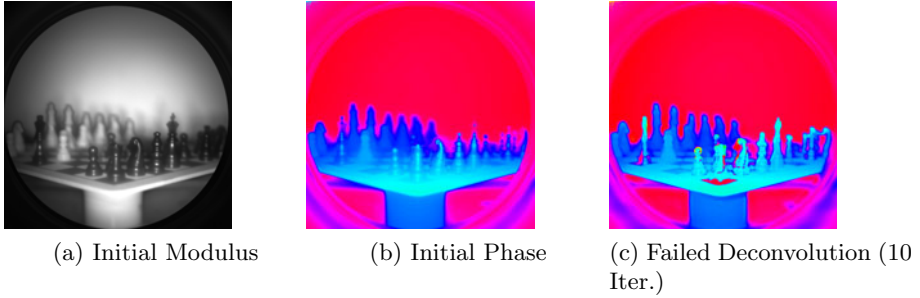


Fig. 10. Scene three, pre- and post-deconvolution. For the phase images, cyan represents objects closer to the camera and red objects farther away. This scene suffers from severe multipath contamination, as shown by the range-intensity coupling for the black chess pieces and squares. A combination of multipath and high phase complexity results in an unsuccessful deconvolution.

scattered from the board in the background resulting in PSF misestimation in addition to having very complicated range content. Since none of the image is saturated, the regions with specular reflections have the most accurate range measurements, which are visibly different from adjacent areas. This is compounded by the fact that each component at a different range within a pixel has a different PSF. Successful restoration of this type of scene awaits a more advanced restoration algorithm that takes into account the range of possible components within each pixel rather than making a naïve assumption that each sample is of an unperturbed single component return.

5 Conclusion

In this paper we have designed a broadband coded-aperture for coding defocus so as to allow depth-of-field to be extended through deconvolution. We have demonstrated that the coded aperture design results in an improvement in restoration performance over a circular aperture and incorporated the coded aperture design into a real full-field AMCW lidar system. The range variation of the defocus and scattering PSFs was sampled and reflection off the image intensifier was isolated as a significant contributor to scattered light. A naïve, proof-of-concept restoration algorithm was demonstrated to substantially improve the quality of some, but not all range-images captured using this new system – difficulties including misestimation of the restoration PSF due to multipath and the naïve assumption of a single component return.

Acknowledgement. This research was supported by a Tertiary Education Commission Top Achiever Doctoral Scholarship and the University of Waikato Strategic Investment Fund.

References

1. Levin, A., Fergus, R., Durand, F., Freeman, W.T.: Image and depth from a conventional camera with a coded aperture. In: Proc. SIGGRAPH 2007 (2007)
2. Ng, R., Levoy, M., Bredif, M., Duval, G., Horowitz, M., Hanrahan, P.: Light field photography with a hand-held plenoptic camera. Technical Report CTSR 2005-02, Stanford University (April 2005)
3. Pentland, A.P.: A new sense for depth of field. *IEEE Trans. Pattern Anal. Mach. Intell.* 9, 523–531 (1987)
4. Chaudhuri, S., Rajagopalan, A.N.: Depth from defocus: a real aperture imaging approach. Springer, Heidelberg (1999)
5. Raskar, R., Agrawal, A., Tumblin, J.: Coded exposure photography: motion deblurring using fluttered shutter. *ACM Trans. Graph* 25, 795–804 (2006)
6. Lumsdaine, A., Georgiev, T.: Full resolution lightfield rendering. Technical report, Adobe Systems Inc. (2008)
7. Dowski, E.R., Johnson, G.E.: Wavefront coding: a modern method of achieving high-performance and/or low-cost imaging systems. In: Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 3779, pp. 137–145 (1999)
8. Mure-Dubois, J., Hugli, H.: Optimized scattering compensation for time-of-flight camera. In: Two- and Three-Dimensional Methods for Inspection and Metrology, Boston, MA, USA, vol. 6762. SPIE, CA (2007)
9. Kavli, T., Kirkhus, T., Thielemann, J.T., Jagielski, B.: Modelling and compensating measurement errors caused by scattering in time-of-flight cameras. In: Huang, P.S., Yoshizawa, T., Harding, K.G. (eds.) Two- and Three-Dimensional Methods for Inspection and Metrology VI, vol. 7066. SPIE, CA (2008)
10. Godbaz, J.P., Cree, M.J., Dorrington, A.A.: Blind deconvolution of depth-of-field limited full-field lidar data by determination of focal parameters. In: Proc. SPIE Computational Imaging VIII, San Jose, California, vol. 7533 (2010)
11. Dorrington, A.A., Cree, M.J., Payne, A.D., Conroy, R.M., Carnegie, D.A.: Achieving sub-millimetre precision with a solid-state full-field heterodyning range imaging camera. *Meas. Sci. and Tech.* 18, 2809–2816 (2007)
12. Oggier, T., Lehmann, M., Kaufmann, R., Schweizer, M., Richter, M., Metzler, P., Lang, G., Lustenberger, F., Blanc, N.: An all-solid-state optical range camera for 3D real-time imaging with sub-centimeter depth resolution (SwissRanger). In: Optical Design and Engineering, St. Etienne, France, vol. 5249, pp. 534–545. SPIE, CA (2004)
13. Larkins, R.L., Cree, M.J., Dorrington, A.A., Godbaz, J.P.: Surface projection for mixed pixel correction. In: Image and Vision Computing New Zealand (IVCNZ 2009), Wellington, New Zealand (2009)
14. Godbaz, J.P., Cree, M.J., Dorrington, A.A.: Multiple return separation for a full-field ranger via continuous waveform modelling. In: Proc. SPIE Image Processing: Machine Vision Applications II, San Jose, California, vol. 7251 (2009)
15. Godbaz, J.P., Cree, M.J., Dorrington, A.A.: Mixed pixel return separation for a full-field ranger. In: Proc. IVCNZ 2008, Christchurch, New Zealand (2008)
16. Marcia, R.F., Harmany, Z.T., Willett, R.M.: Compressive coded apertures for high-resolution imaging. In: Optics, Photonics, and Digital Technologies for Multimedia Applications, vol. 7723. SPIE, CA (2010)
17. Landweber, L.: An iteration formula for Fredholm integral equations of the first kind. *Am. J. of Math.* 73, 615–624 (1951)

Surface Extraction from Iso-disparity Contours

Chris McCarthy¹ and Nick Barnes^{1,2}

¹ NICTA Canberra Research Lab
Canberra, Australia

² The Australian National University
College of Engineering and Computer Science,
Canberra, Australia

Abstract. This paper examines the relationship between iso-disparity contours in stereo disparity space and planar surfaces in the scene. We specify constraints that may be exploited to group iso-disparity contours belonging to the same planar surface, and identify discontinuities between planar surfaces. We demonstrate the use of such constraints for planar surface extraction, particularly where the boundaries between surfaces are orientation discontinuities rather than depth discontinuities (*e.g.*, segmenting obstacles and walls from a ground plane). We demonstrate the advantages of our approach over a range of indoor and outdoor stereo images, and show that iso-disparity analysis can provide a robust and efficient means of segmenting smooth surfaces, and obtaining planar surface models.

1 Introduction

The extraction of rigid surfaces from stereo images or camera motion has been a topic of interest in computer vision for many years. In structured and semi-structured environments, extracting planes or locally planar continuous surfaces provides a basis for a range of tasks including 3D reconstruction [1] and obstacle detection/avoidance [2].

Particular focus has been given to surface/plane segmentation in 2D disparity space. Oh *et al.* [3] use plane fitting over initial point matches within colour segmented regions to refine disparity estimates. Hong and Chen [4] combine a similar framework with graph cuts to refine stereo matching and extract planar surface segments. Other graph-cut based disparity labelling examples include [5,6]. Se and Brady [7] apply random sample consensus (RANSAC [8]) to identify the assumed dominant ground plane. Thakoor [9] segment planar surfaces by alternating between disparity segmentation using local surface smoothness models until convergence. Other techniques such as [10,2] acquire planar surface models via Hough-based voting over disparities back projected into Euclidean space. While applying RANSAC or Hough-based voting over disparities provides robustness to outliers, both are reliant on a reasonable inlier set.

The central determinant of surface appearance in disparity space is how it interacts with iso-disparity surfaces of the stereo configuration. For any given

disparity, there exists an *iso-disparity surface* in \mathbb{R}^3 that projects into both image planes with uniform displacement along epipolar lines (*i.e.*, the associated disparity). Physical surfaces intersect iso-disparity surfaces, forming *iso-disparity contours*. These contours are determined by both the geometry of the surface, and the stereo configuration used to image it. While significant attention has been given to the properties of iso-disparity curves [11] (and related iso-motion curves [12]) with respect to relative camera poses, less consideration has been given to their possible use for inferring surface geometry. In stereo segmentation, less attention has been paid to segmenting surfaces around orientation discontinuities rather than depth discontinuities.

In this paper we examine constraints on iso-disparity contours across projected planar surfaces in the scene. We demonstrate the application of these constraints to planar surface segmentation in disparity space. We consider these constraints for general stereo configurations, however, we focus predominantly on the rectified parallel stereo case.

The remainder of the paper is structured as follows. Section 2 reviews stereo disparity and iso-disparity surfaces. Section 3 presents a formulation of iso-disparity contours for parallel rectified stereo, and specifies constraints on contours across planar surfaces, and planar surface boundaries. Section 4 provides an implementation overview of the proposed iso-disparity segmentation technique. Section 5 presents experimental results for surface segmentation and modelling tasks over indoor and outdoor disparity images. Section 6 concludes the paper.

2 Background

2.1 Stereo Disparity

Stereo disparity is the measure of the distance between the projected location of a single point in two views. Let \mathbf{P} and \mathbf{P}' denote two camera matrices, and X , a point in space projecting into both views such that:

$$x = \mathbf{P}X, \quad x' = \mathbf{P}'X,$$

where x and x' are the projected location of X in the left and right image planes respectively.

Under a general stereo configuration, the fundamental matrix, \mathbf{F} , defines a mapping of points in one image to corresponding epipolar lines in the other such that:

$$l' = \mathbf{F}x, \tag{1}$$

where l' denotes the epipolar line [13]. The relative change of position between x and x' will occur along l' .

Pollefeys and Sinha [11] propose measuring disparity as a scalar distance along epipolar lines such that:

$$d = \lambda'(x') - \eta\lambda(x), \tag{2}$$

where $\lambda(x) = |x - e| - l_o$, and l_o is the distance between the epipole, e , and its closest point in the image. $\eta \in -1, 1$ represents the change of sign required when epipolar line orientations differ.

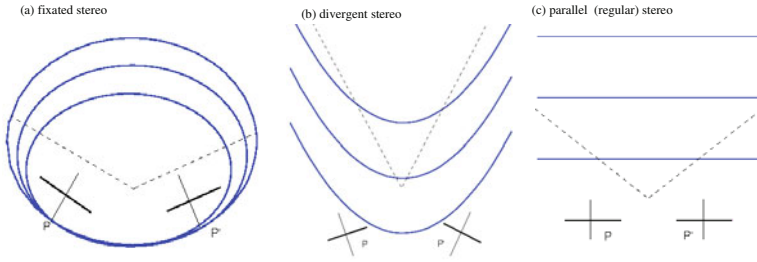


Fig. 1. Iso-disparity curves for different stereo configurations

2.2 Iso-disparity Surfaces

Within each epipolar plane there exists a family of iso-disparity conics passing through both camera centres [11]. Points on any given conic project with uniform disparity along corresponding epipolar lines. Each conic is distinguished by the level of disparity it represents, increasing monotonically with proximity to the cameras. The set of iso-disparity conics for a given disparity level represent a series of intersections of the associated iso-disparity surface in \mathbb{R}^3 , with each epipolar plane.

The form of conic sections in the epipolar plane is dependent on the relative camera configuration. Figure 1 shows example curves of each case. A well known degenerate case is when camera optical axes are parallel (and non co-linear). Iso-disparity surfaces become a series of positively signed iso-disparity planes with the zero iso-disparity plane moving to the infinite plane. The physical depth, Z , of each plane is given by the well known equation:

$$Z = -\frac{fb}{d}, \tag{3}$$

where d is the disparity, f is the focal length, b is the stereo baseline.

Surfaces in the scene become apparent in disparity space through their intersection with iso-disparity surfaces. We refer to these intersections as iso-disparity contours.

3 Extracting Planes from Iso-disparity Contours

Iso-disparity contours define a relationship between the stereo configuration and surfaces in the scene. We specify constraints on this relationship that may be exploited to detect and segment planar surfaces, and identify discontinuities between connected planar surfaces in disparity space. We consider these in the context of rectified parallel stereo cameras, however these properties may be extended to any stereo configuration where iso-disparity surfaces are continuous.

3.1 Formulation of Iso-disparity Contours

Let π_d be a single iso-disparity plane, where $d \in [0, \infty]$ is the associated disparity of the plane. Consider a continuous surface, $Q \in \mathbb{R}^3$, occupying some portion

of the overlapping visual field of both cameras. The appearance of Q in stereo disparity can be described as an ordered set of curves, \mathcal{S} representing the intersections of Q with each iso-disparity plane occupying the same depth. Let $C_d(t) \in \mathbb{R}^2$ represent the curve of intersection of Q with π_d , where $t \in \mathbb{R}$ is a parameterised distance along the curve. Thus:

$$\mathcal{S} = \left\{ C_i(t) : i \in [d_{\min}, d_{\max}] \right\}, \quad (4)$$

where d_{\min} and d_{\max} are the disparity extrema of the surface, and i is continuous between d_{\min} and d_{\max} . We discuss the special case of a surface not varying in depth (*i.e.*, fronto-parallel) at the end of this section.

Let $D(x') \in \mathbb{R}$ be a disparity value at location $x' \in \mathbb{R}^2$ in the disparity image D . Assuming D is aligned with a reference image frame, we apply the appropriate camera matrix to points along $C_d(t)$ such that:

$$c'_d(t) = \mathbf{P}C_d(t), \quad (5)$$

where $c'_d(t) \in D$. Under pinhole projection, all gradients of $C_d(t)$ (which lie within a fronto-parallel plane to D) will be preserved up to scale, and thus $c'_d(t)$ will retain the same form as $C_d(t)$. We define the projection of \mathcal{S} into D as:

$$\mathcal{S}' = \left\{ c'_i(t) : i \in [d_{\min}, d_{\max}] \right\}. \quad (6)$$

Note that all labels defined in D are marked with an apostrophe.

Let Q be a planar surface with surface normal \hat{n} . In this case, all $C_d(t) \in \mathcal{S}$ will be linear and parallel, forming an ordered set of monotonically increasing/decreasing iso-disparity surface intersections. Thus, any point of interest $x'_o \in Q$ in the field of view will form part of a linear iso-disparity contour in disparity space such that:

$$c'_d(t) = \mathbf{P}(t(\hat{n} \times \hat{z}) + x'_o), \quad (7)$$

where $\hat{z} \in R^3$ is the normal to the intersecting iso-disparity plane (*i.e.*, the direction of parallel optical axes) and t is a parameterised distance along the line. Figure 2 shows an example scene depicting an environment of predominantly connected surfaces and the iso-disparity contours across these surfaces.

3.2 Determining Planes along Iso-disparity Contours

We now consider the task of inferring unknown planar surfaces from iso-disparity contours. In continuous disparity space, any interest point x'_o projecting from a depth varying surface will form part of an iso-disparity contour. We may describe this contour as a level set of adjoining points such that:

$$\mathcal{C}'(x'_o) = \left\{ x' : D(x') = D(x'_o) \right\}. \quad (8)$$

Note that \mathcal{C}' is not limited to a single surface as it may traverse a series of connected surfaces in the scene.

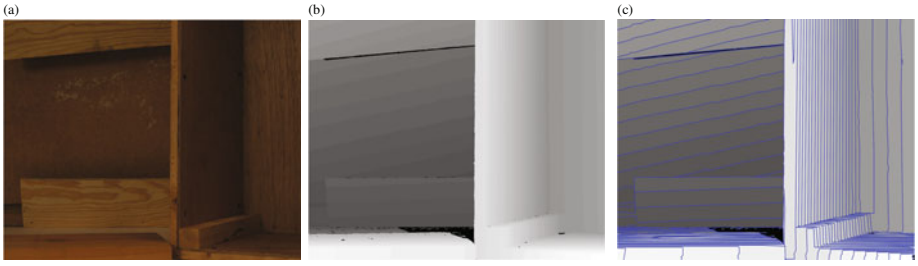


Fig. 2. ‘wood’ from the 2006 Middlebury data set depicting adjoining planar surfaces with (a) the original image, (b) the resulting ground truth disparity map, and (c) the disparity map with iso-disparity contours showing the piece-wise linear segments that form the total contour for each level set

If x'_o lies on a plane (or piece-wise planar surface), then neighbouring points in \mathcal{C}' must locally fall along a line passing through x'_o . Let $f(t)$ be a parametrised vector function defining a line of points in \mathcal{C}' such that: $f(0) = x'_o$. Thus we have:

$$D(f(0)) = D(x'_o) = d, \quad (9)$$

where d is a scalar disparity value. Differentiating (9) at x'_o via the chain rule we obtain:

$$\nabla D(x'_o) \cdot \nabla f(0) = 0, \quad (10)$$

where $\nabla(\cdot)$ defines the gradient vector.

Given $f(t)$ defines a straight line in \mathcal{C}' , it follows that:

$$\forall t, \nabla f(t) = \nabla f(0), \quad (11)$$

and thus:

$$\nabla D(f(t)) \cdot \nabla f(t) = 0. \quad (12)$$

That is, $\nabla D(f(t))$ and $\nabla f(t)$ must be perpendicular. The above also implies that:

$$\forall t, \nabla D(f(0)) = \nabla D(f(t)), \quad (13)$$

Thus, within a planar surface region in disparity space, both $\nabla D(f(t))$ and $\nabla f(t)$ are also constant. By testing for conformity with Constraints 12 and 13, co-planar groupings of points may be defined along iso-disparity contours.

3.3 Determining Planes along Disparity Gradients

To uniquely determine the plane, co-planar points not in $f(t)$ must also be determined (*i.e.*, on other iso-disparity contours). For this we consider the orthogonal direction as given by $\widehat{\nabla D}$, the unit vector in the direction of ∇D . As in the iso-disparity case, co-planar points will lie along a straight line. Following a similar derivation as in the iso-disparity case, we define a straight line, $g(s)$, of points in the direction of ∇D , such that:

$$g(0) = x'_o, \quad (14)$$

where x'_o is a point of interest, and by definition:

$$\widehat{\nabla g(0)} = \widehat{\nabla D}(g(0)). \tag{15}$$

Thus:

$$\forall s, \nabla g(0) = \nabla g(s), \tag{16}$$

and,

$$\widehat{\nabla D}(g(s)) \cdot \widehat{\nabla g}(s) = 1, \tag{17}$$

That is, all co-planar points with x'_o in the direction $\nabla D(x'_o)$ will lie along a straight line (16), and will always be in the direction of $\nabla D(g(s))$ (17).

Unlike iso-disparity contours, disparity values will vary along $g(s)$. We note that across a plane, disparity varies linearly. Thus, for a point $x'_n \in g(s)$ to be co-planar with x'_o , it must satisfy:

$$\nabla D(x'_n) = \nabla D(x'_o). \tag{18}$$

To enforce the linear constraint on the disparities themselves, we define:

$$D(x'_n) = D(x'_o) + \nabla D(x'_o) \|x'_n - x'_o\|. \tag{19}$$

To summarise, co-planar points must exhibit a constant direction and magnitude of maximum disparity change (18), and measured disparities at each point must adhere to a linear model. Given an initial grouping of points along iso-disparity contours is performed, the grouping of co-planar points in \mathcal{G}' (via Constraints (18) and (19)) is equivalent to grouping co-planar iso-disparity contours.

3.4 Determining Surface Orientation Boundaries

For segmentation, it may be preferable to extract points of orientation discontinuity, allowing for more explicit representations of surface boundaries. In the absence of noise, points of non-conformity with planar constraints along iso-disparity contours (and disparity gradients) will only occur where a surface boundary exists in the scene. Thus, we may discover surface boundaries by locating points of discontinuity along these curves.

Consider an iso-disparity contour $\mathcal{C}'(x'_o)$. Following directly from Constraint (12), an orientation discontinuity with respect to a point of interest, x'_o , may be defined as any point, $x_d \in \mathcal{C}'(x'_o)$ that satisfies:

$$|\widehat{\nabla D}(x'_d) \cdot \widehat{\nabla f}(0)| > e, \tag{20}$$

where $\widehat{\nabla f}(0)$ gives the direction of the line of points $f(t) \in \mathcal{C}'(x'_o)$ passing through x'_o ($f(0) = x'_o$), and e is a discontinuity threshold. We use only the directions of both gradient vectors as we are concerned only with their relative orientations.

Let $\mathcal{G}'(x'_o)$ define an orthogonal set of points to $\mathcal{C}'(x'_o)$ at x'_o . A point $x'_n \in \mathcal{G}'(x'_o)$ represents a surface orientation discontinuity with respect to x'_o if one or both of the following are satisfied:

$$\left| 1 - (\widehat{\nabla D}(x'_n) \cdot \widehat{\nabla D}(x'_o)) \right| > e, \tag{21}$$

or,

$$\left| \frac{D(x'_n) - D(x'_o)}{\|x'_n - x'_o\|} - \nabla D(x'_o) \right| > \epsilon, \quad (22)$$

That is, where the disparity gradient vectors are not parallel, or where conformity with a linear model of disparity is broken (*i.e.* the converse of . Both constraints represent the converse of Constraints [18](#) and [19](#)).

We have described the above properties in continuous disparity space. However, in practise, disparities are typically sampled from a finite set of values. The analysis of disparities along gradients must account for this. Specifically, $\nabla D(x')$ and $\|\nabla D(x)\|$ will be locally measurable only at points along iso-disparity contours (where a transition between disparity levels exists). Thus, a search through a separating iso-disparity region is required to connect co-planar contours.

3.5 Special Cases and Exceptions

Two special cases exist: 1. Q is a plane fronto-parallel to the image plane, thus generating no iso-disparity contours; and 2. Q is entirely within an epipolar plane, whereby all intersections with π_d project to the same epipolar line in D . The former case is easily detected as a large uninterrupted area of constant disparity. The latter case is uncommon given sufficient image resolution. Further, most objects have finite volume.

4 Implementation Overview

We now application of the iso-disparity constraints for planar surface extraction in disparity space. We now describe how the iso-disparity constraints outlined above may be used to determine the number and location of planar surfaces in the scene.

4.1 Extracting and Segmenting Iso-disparity Contours

We apply a standard level sets framework (see [14](#) for a review). For each discrete disparity level, d , an initial signed distance function, $\phi(x') : x' \in D$, is defined for all image points (distance is defined as the Euclidean distance to the closest set boundary). Points with disparity equal to d are assigned negative values, while all others positive. We then fit a level set with a curvature constraint included to provide robustness to non-genuine discontinuities.

Each contour set is broken into linear segments (determined by a maximum allowable distance of points from the line), thereby enforcing Constraints [12](#) and [13](#) within each segment. Adjacent line segments sufficiently close in angle (*i.e.*, Constraint [20](#) is not satisfied) are grouped as one linear iso-disparity contour segment. Otherwise, the point is marked as an orientation discontinuity.

4.2 Iso-disparity Contour Grouping

Surface groups are formed by searching along maximum disparity gradients (*i.e.*, $\nabla D(x')$) of each linear iso-disparity segment. A flood fill style search is

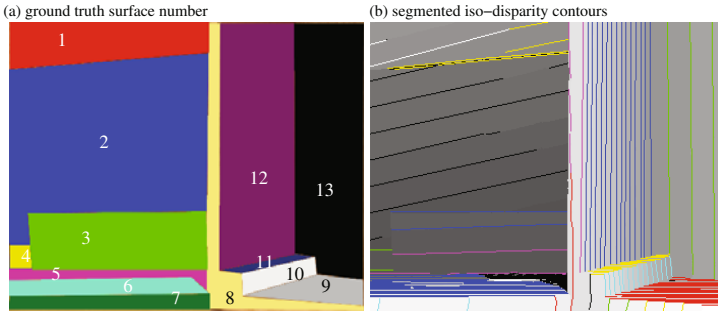


Fig. 3. Planar surface extraction results for ‘wood’ from the Middlebury 2006 data set, showing (a) a hand segmented image showing the number of planar surfaces in the scene (ground truth = 13), and (b) results from the segmentation of iso-disparity contours into planar surface regions. All 13 surfaces are successfully detected and separated along orientation and depth boundaries. Some over segmentation within planar regions caused $\frac{5}{13}$ false positive surface separations to be recorded.

applied to find connections between iso-disparity linear segments from an initial seed selected from the set of all linear segments, \mathcal{L} . The search for connecting line segments is performed above and below the seed line. Valid connections are determined through their conformity with Constraints [18](#) and [19](#), and are added to a surface set, \mathcal{S} . Where a linear segment represents a non-smooth transition between disparity levels it is labelled as a depth discontinuity and the search is terminated along that branch. The procedure is recursively applied to all newly found connecting linear segments until all possible connections are exhausted. The set of assigned lines \mathcal{S}' for the surface are then removed from \mathcal{L} , and a new surface group seed is selected from those remaining in \mathcal{L} , until \mathcal{L} is empty.

5 Experimental Results

Experiments were conducted to examine the iso-disparity contour constraints for surface segmentation and modelling tasks. The experiments presented are motivated by the primary application area of this work in delivering visual navigation assistance to the vision impaired.

5.1 Planar Surface Extraction

The iso-disparity algorithm was tested over high quality disparities from the Middlebury ‘wood’ test image pairs [15](#). The images [Figure 2\(a\)](#) depict surface boundaries defined by an orientation rather than depth discontinuities. [Figure 3\(a\)](#) shows a ground truth (hand) segmentation of the scene. There are 13 planar surfaces in the scene, each distinguished from each other. [Figure 3\(b\)](#) shows the result of the iso-disparity segmentation algorithm, using colour coding to distinguish different surface groupings. All surfaces are successfully separated along orientation and depth boundaries. Over segmentation is apparent within

near fronto-parallel planar regions, resulting in $\frac{5}{13}$ false positive separations. No false positives are recorded within regions with significant depth variation.

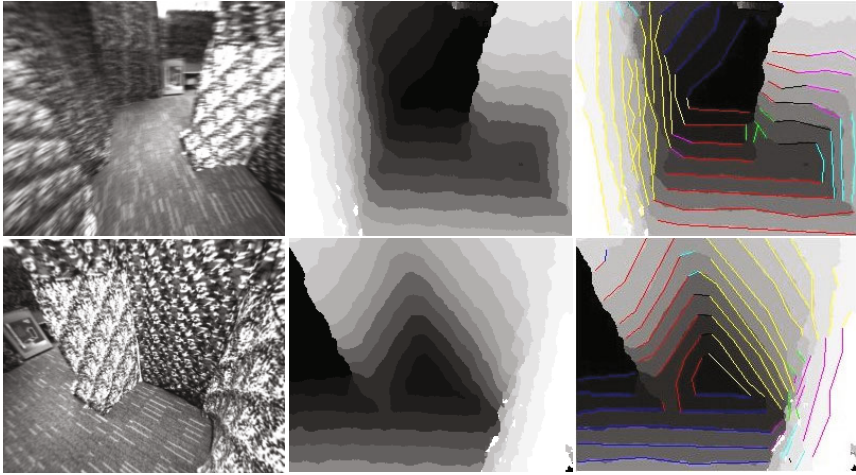


Fig. 4. Results for planar surface segmentation. Left column shows original (left) images from the stereo camera, middle column the histogram equalised disparity images, and right column, the disparity images with segmented iso-disparity contours overlaid. Different colours represent separate segmented regions.

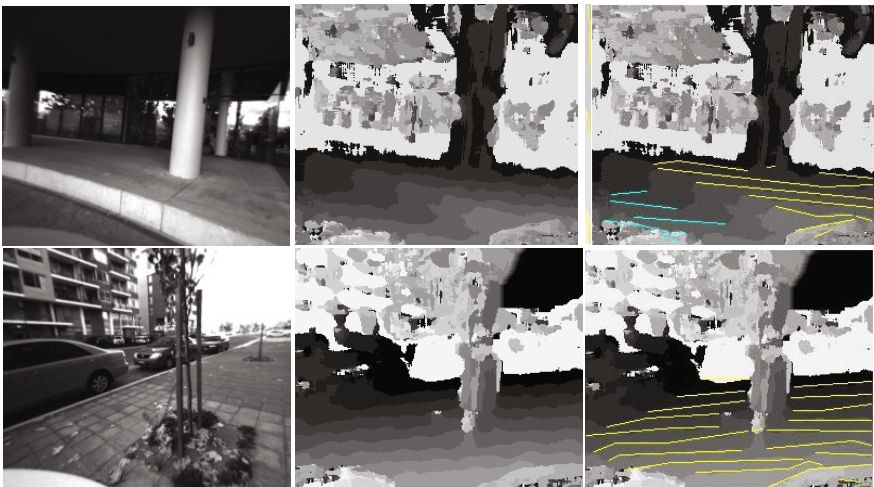


Fig. 5. Ground plane segmentation results. Left column shows images from the stereo camera, middle column the disparity images, and right column, the disparity images with segmented ground plane iso-disparity contours overlaid. Different colours represent separate segmented regions closely matching the dominant plane (in yellow).

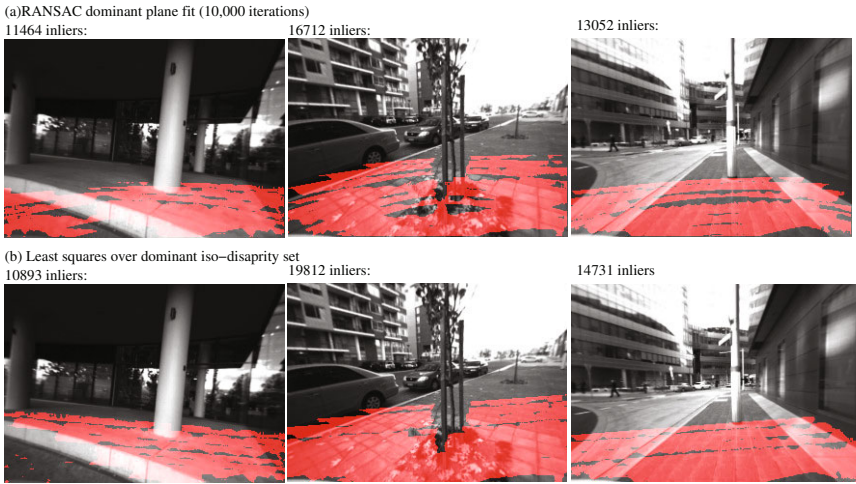


Fig. 6. Inlier sets for plane fitting using (a) standard RANSAC (10,000 iterations), and (b) Least squares over iso-disparity segmentation

5.2 Surface Segmentation during Navigation

Experiments were conducted to assess performance in determining traversable space in indoor and outdoor environments, and with disparity maps of less quality. Figure 4 shows results of the iso-disparity contour surface segmentation for two sample images taken from our own test environment. This environment consists of a heavily textured floor and a series of planar curtain walls. Disparity images were obtained from a stereo rig¹ at points within the environment. Different colours indicate disjoint planar sections. Results show the segmentation of all iso-disparity contours for all dominant locally planar surfaces. Some over-segmentation is apparent in the top row results (the right planar surface). However, all major surfaces are successfully detected and segmented.

5.3 Ground Surface Extraction

A stereo rig was walked through a typical urban outdoor environment and raw disparities recorded. Figure 5 shows sample frames, and corresponding disparity images from the sequence. The ground surface was assumed to be the dominant surface in the image (*i.e.*, the surface generating the most iso-disparity contours), and is shown as yellow contours in the results. Results show the dominant ground plane region is successfully identified in each image.

Least squares plane fitting was also applied over disparities along segmented ground plane contours. A standard dominant plane RANSAC implementation was also tested for comparison. Figure 6 shows the resulting inlier sets (in red) for the extracted best fit planar models for each sample. The top row shows results for traditional RANSAC (after 10,000 iterations), and below for least

¹ Point Grey Research IEEE 1394 Bumblebee2 stereo camera.

squares over segmented iso-disparity contours. Results show accurate ground plane models are obtained from a least squares fit over segmented disparity contours. Inliers from the iso-disparity model appear more accurate, and more abundant within correct regions. Surface discontinuities also appear to be better preserved using the iso-disparity inlier sets. This is most evident in the left image, where a clear distinction between the upper and lower surface is evident. In contrast, RANSAC attempts to fit a plane across both surfaces, yielding greater inlier support, but a less accurate model in the context of identifying obstacles such as the step.

6 Conclusion

We have examined the relationship between iso-disparity surfaces and the geometry of planar surfaces in the scene. We have exposed properties of iso-disparity contours across planar surfaces, and at surface boundaries marked by a change of orientation. From this, we have demonstrated the application of these properties for a range of surface segmentation and modelling tasks.

Acknowledgements. NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

References

1. Bartoli, A.: A random sampling strategy for piecewise planar scene segmentation. *Computer Vision and Image Understanding: CVIU* 105, 42–59 (2007)
2. Okada, K., Kagami, S., Inaba, M., Inoue, H.: Plane segment finder: algorithm, implementation and applications. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, pp. 2120–2125 (2001)
3. Oh, J.D., Ma, S., Kuo, C.C.: Stereo matching via disparity estimation and surface modeling. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8 (2007)
4. Hong, L., Chen, G.: Segment-based stereo matching using graph cuts. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. I-74–I-81 (2004)
5. Kolmogorov, V., Zabih, R.: Computing visual correspondence with occlusions using graph cuts. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, vol. 2, pp. 508–515 (2001)
6. Sinha, S.N., Steedly, D., Szeliski, R.: Piecewise planar stereo for image-based rendering. In: *Proceedings of the IEEE International Conference on Computer Vision, ICCV* (2009)
7. Se, S., Brady, M.: Stereo vision-based obstacle detection for partially sighted people. In: Chin, R., Pong, T.-C. (eds.) *ACCV 1998. LNCS*, vol. 1352. Springer, Heidelberg (1997)
8. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography. *Communications of the ACM* 24(6), 381–395 (1981)

9. Thakoor, N., Jung, S., Gao, J.: Real-time planar surface segmentation in disparity space. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8 (2007)
10. Trucco, E., Isgro, F., Bracchi, F.: Plane detection in disparity space. In: International Conference on Visual Information Engineering, VIE 2003, pp. 73–76 (2003)
11. Pollefeys, M., Sinha, S.N.: Iso-disparity surfaces for general stereo configurations. In: Pajdla, T., Matas, J.(G.) (eds.) ECCV 2004. LNCS, vol. 3023, pp. 509–520. Springer, Heidelberg (2004)
12. Fermüller, C., Aloimonos, Y.: On the geometry of visual correspondence. *International Journal of Computer Vision* 21, 223–247 (1997)
13. Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*, 2nd edn. Cambridge University Press, Cambridge (2000)
14. Cremers, D., Rousson, M., Deriche, R.: A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape. *International Journal of Computer Vision* 72, 215 (2007)
15. Hirschmiller, H., Scharstein, D.: Evaluation of cost functions for stereo matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2007)

Image De-fencing Revisited

Minwoo Park¹, Kyle Brocklehurst¹, Robert T. Collins¹, and Yanxi Liu^{1,2}

¹ Dept. of Computer Science and Engineering

² Dept. of Electrical Engineering

The Pennsylvania State University, University Park, PA,16802, USA

{mipark,brockleh,rcollins,yanxi}@cse.psu.edu

Abstract. We introduce a novel image defencing method suitable for consumer photography, where plausible results must be achieved under common camera settings. First, detection of lattices with see-through texels is performed in an iterative process using online learning and classification from intermediate results to aid subsequent detection. Then, segmentation of the foreground is performed using accumulated statistics from all lattice points. Next, multi-view inpainting is performed to fill in occluded areas with information from shifted views where parts of the occluded regions may be visible. For regions occluded in all views, we use novel symmetry-augmented inpainting, which combines traditional texture synthesis with an increased pool of candidate patches found by simulating bilateral symmetry patterns from the source image. The results show the effectiveness of our proposed method.

1 Introduction

We address a real-life problem in photo editing where one would like to remove or change fence-like, near-regular foreground patterns that are often unavoidable, as illustrated in Figure 1. This task was first addressed by Liu et al. [1] by a 3-step procedure 1) lattice detection [2], 2) foreground / background segmentation and 3) inpainting [3,4]. Lattice detection and foreground/background segmentation in [1] proceeded sequentially, hence an abundant amount of information arising from the repeating pattern was not fully utilized. Furthermore, the performance of previous lattice detection algorithms [2] is far from practical for this application due to inaccuracy and slowness.

In this paper, we make the following novel contributions to this challenging goal; **• online learning and classification** is used to aid lattice detection and segmentation, resulting in a substantial improvement in detection rate over current state-of-the-art lattice detection algorithms [5,2]. Our online classification and segmentation method is not confined to this specific application; it can be applied to other near-regular texture detection and analysis tasks. **• multiview inpainting** is introduced to improve the region filling process by using multiple, shifted camera views, since the best way to infer an unknown pixel is to see the occluded region in another view. The approach does not assume any rigidity of the fence nor objects, but requires some offset between views: either by camera or object movement. These are practical requirements for every-day photography, since one can



Fig. 1. (a) Input image (b) Automatic segmentation using online learning (c) Result of Liu et al. [1] (d) Result of our proposed method

take multiple photos of a scene simply by shifting the camera, revealing objects behind the fence due to parallax. • **symmetry-augmented inpainting** is introduced to tackle the problem of scarcity of candidate samples after large amounts of foreground have been removed leaving fragmented background pixels. We increase the candidate pool by simulating bilaterally symmetric patches from the source image. For instance, if half of someone’s mouth is covered, we can recover the occluded region reliably from the opposite side of the mouth by reflecting that patch. The experimental results show the effectiveness of our proposed method, especially for objects that are extremely unforgiving to flawed inpainting such as a human face and structured backgrounds (see Figures [1] and [8] for examples).

2 Related Work

Liu et al. [1] introduce a novel application in computational photography by taking advantage of see-through NRTs to remove a near regular foreground. As the authors of [1] point out, each of the components in the application is very challenging on its own and poses many research questions.

2.1 Lattice Detection

There is a rich body of work on lattice detection in the literature [6, 2, 5, 7, 8, 9, 10, 11]. However, it was Hays et al. [2] who first developed an automatic deformed lattice detection algorithm for real images without pre-segmentation. The method of [2] is based on looking for the neighbors of a randomly selected interest

point in the image. If a sufficient number of points look like their respective t_1, t_2 neighbors (lower order similarity) and also share their t_1, t_2 neighbors' directions/orientations (higher order correspondences) towards other interest points in the image, those points and their neighborhood relationships are confirmed to be part of the lattice. Based on this partial result, the slightly deformed lattice is straightened out and a new round of lattice discovery starts, so the extracted lattice grows bigger and bigger. Formulating the lattice detection problem as a higher order correspondence problem adds computational robustness against geometric distortions and photometric artifacts in real images, and the publicly available code produces impressive results.

Later, Park et al. [5] developed a deformed lattice detector within a Markov Random Field using an efficient inference engine called Mean-Shift Belief Propagation. They showed 72% improvement in lattice detection rate over the Hays' algorithm [2], with a factor of 10 speed up.

However, all algorithms discussed so far ignore the foreground/background characteristics of the repeating pattern we want to find. In particular, images which contain fence-like structures are inevitably highly irregular despite the regularity of the foreground. For such cases, the irregular background interferes with the detection of the see-through foreground lattice. Our method learns the type of the repeating pattern, removes the irregularities, and uses the learned regularity in evaluating the foreground appearance likelihood during lattice growth, a crucial improvement since robust and complete lattice detection plays the most significant role in our application.

2.2 Image Completion

Traditional texture filling tools such as Criminisi et al. [3,4] require users to manually mask out unwanted image regions. Based on our own experience, for images such as those in Figures 1, 7, and 8, this process is very tedious and error-prone. Simple color-based segmentations are not sufficient. Painting a mask manually, as in previous inpainting work, requires copious time and attention because of the complex topology of the foreground regions.

Favaro et al. [12] introduce a method for the restoration of images in which certain areas have been blurred. Their method develops a map of the relative amount of blur at each position in the image, then learns correspondences between recurring objects or image patches. This allows them to copy the least blurred occurrence of an object and paste patches from it to inpaint over blurred occurrences of the same or similar objects. This is a powerful method of relating undesirable blur utilizing the power of understanding multiple instances of the same object in a scene. Their work differs from ours in that they do not attempt to use or understand any underlying structure, such as a lattice, that may exist among the instances of the recurring object. Also, their method of inpainting removes blurring, such as that from varying depth, but does not remove occlusion, such as a fence-like foreground region.

As an extension to photo inpainting, Wexler et al. [13] and Patwardhan et al. [14] each propose a video inpainting method. This is desirable, since temporal

information can give additional information that can aid the inpainting process. Although the balance of spatial and temporal continuity is far from trivial, both methods produced spatially and temporally coherent results, albeit at the cost of needing to mask out unwanted regions manually. With these filling tools, a user has the capability to reveal content in a photo behind occlusions. However, if the missing region is part of a complex object with high resolution, such as a human subject, the quality of inpainting is often insufficient, as can be seen in Figure 7 and 8.

Hays and Efros [15] proposed a scene completion method using millions of photographs. The algorithm fills in the hole regions in images with seamless and semantically valid patches from the database. However neither the database images nor the regions to be filled are fragmented by any foreground structures.

Vaish et al. [16] proposed a method to reconstruct densely occluded scenes using synthetic aperture photography. However, they require a large, synchronized camera array (30 ~ 100 cameras) to achieve this goal, which is obviously impractical for consumer-grade use.

Our approach represents a middle ground between traditional image completion and video completion/synthetic aperture reconstruction, since we use only a small number of auxiliary images that are easily achievable in everyday photography.

3 Near Regular Texture Segmentation

Our basic lattice detection algorithm is similar to [5]. The procedure is divided into two phases, where the first phase proposes one (t_1, t_2) -vector pair and one texture element, or texel. 2D lattice theory tells us that every 2D repeating pattern can then be reconstructed by translating this texel along the t_1 and t_2 directions. During phase one, we detect KLT corner features, extract texture around the detected corners, and select the largest group of similar features in terms of normalized correlation similarity. Then we propose the most consistent (t_1, t_2) -vector pair through an iterative process of randomly selecting 3 points to form a (t_1, t_2) pivot for RANSAC and searching for the pivot with the maximum number of inliers.

At phase two, tracking of each lattice point takes place under a 2D Markov Random Field formulation with compatibility functions built from the proposed (t_1, t_2) -vector pair and texel. The lattice grows outwards from the initial texel locations using the (t_1, t_2) -vector pair to detect additional lattice points. The tracking is initiated by predicting lattice points using the proposed (t_1, t_2) -vector pair under the MRF formulation. The inferred locations are further examined; if the image likelihood at a location is high, then that location becomes part of the lattice. However, for robustness, the method avoids setting a hard threshold and uses the region of dominance idea introduced in [6]. This is particularly important since there is no prior information about how many points to expect in any given image. If the threshold of detecting lattice points is too high, then recall rate suffers.

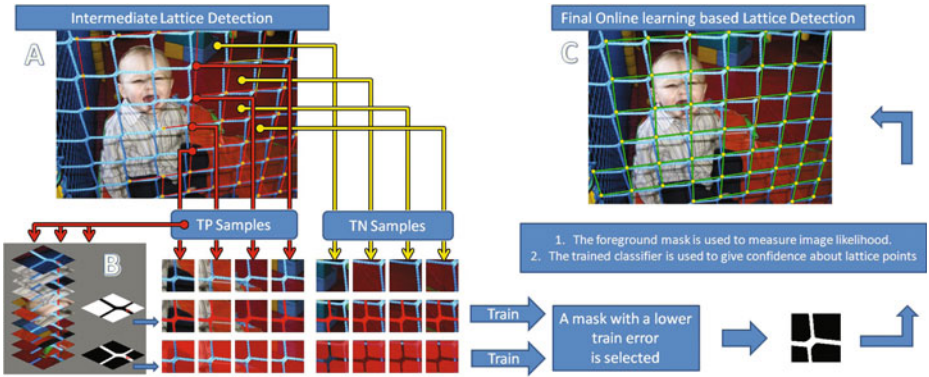


Fig. 2. Procedure of lattice detection using online clustering, learning and classification

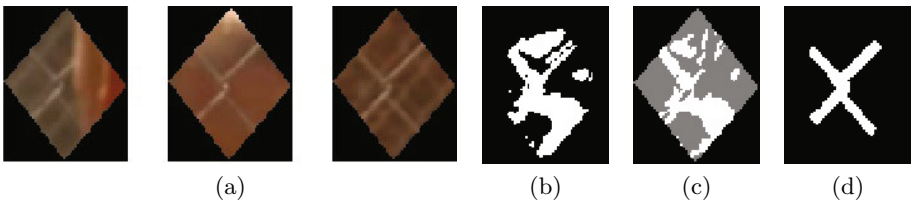


Fig. 3. Sample FG/BG classification for a layer mask. (a) sample texels from the lattice are shown. (b)-(c) results of two methods proposed in [11]. (d) results of our proposed method.

Since the performance of lattice detection plays an essential role in this application, we introduce a better decision system that uses online classification and combines the lattice detection procedure with foreground / background segmentation. In addition, we segment out the foreground layer during the detection procedure and build a mask to remove noisy regions of each texel to represent background irregularities from distracting and misleading the inference procedure. Since evaluation of a noisy image likelihood could misdirect the inference of new texel locations, resulting in inaccurate lattice detection, we evaluate the image likelihood of the each texel by normalized cross correlation using only the foreground mask.

3.1 Clustering for the Foreground Segmentation

Liu et al. [11] simultaneously align multiple texels by calculating a homography for each texel that brings its corners into alignment with the average texel shape (Figure 2B). After aligning all the texels, they compute the standard deviation of each pixel in each texel with respect to the values at the same location in all other texels. They propose two methods of pixel classification. The first was the classification of background versus foreground by thresholding of the variance among corresponding pixels. The second was to consider the color of each texel along with

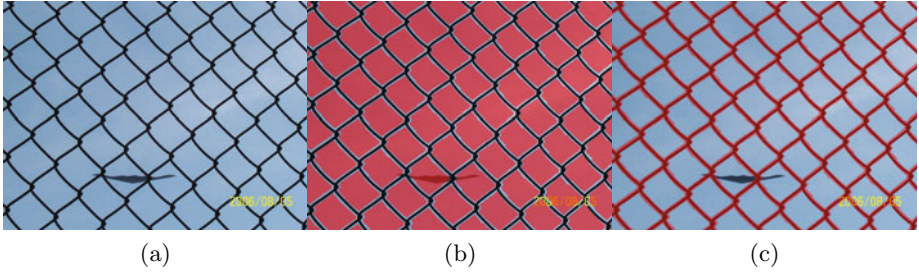


Fig. 4. (a) An uniform background can make the relative mean RGB variance of the foreground larger. (b) Results of taking the cluster with lower variance as foreground: red is foreground. (This picture is best viewed in color.) (c) Results of our proposed foreground segmentation.

the variance and performing K-means clustering on 6D vectors composed of the value and standard deviation of red, green, and blue channels for each pixel. They identified the pixels belonging to the lower variance cluster as the lattice region. Sample results from these two methods are shown in Figures 3b and 3c.

Differing from Liu et al. [1], we use the mean of all pixels at each location within the average texel shape. Now the input to the K-means ($K=2$) clustering is a set of 6D vectors composed of the mean value (for all pixels at that location) and the standard deviation (for each pixel) of red, green, and blue channels. We achieve better results with the use of the mean value, as can be seen in Figure 3d. This is because the means cancel out the irregularities in the backgrounds and make the boundary between the foreground and the background clear.

However, taking the cluster with a smaller RGB variance does not always work since severe lighting conditions on the foreground or a uniform background can result in equivalent RGB variance for each cluster, as can be seen in Figure 4.

3.2 Online Learning-Based Lattice Detection

In our lattice detection algorithm, online learning using a support vector machine is performed to improve the classification of lattice points and for foreground segmentation. The base lattice detection algorithm provides both samples, $x_i \in R^n$ and the label of the samples $y_i = \{-1, 1\}$, which enables us to do supervised learning. Positive samples, $x_i, y_i = 1$ are collected from patches centered at lattice points (Figure 2, red arrows). Negative samples $x_i, y_i = -1$ are collected from patch locations between positive samples (Figure 2, yellow arrows). Next, we segment the lattice region to determine the lattice mask using K-means (Section 3.1 and Figure 2 B). At this stage we have two candidates for the foreground mask. Then, at each sample location, RGB color histograms are computed from the two masks and used as features.

We use a support vector machine (SVM) with linear kernel and 10-fold cross validation. We train the SVM to minimize the objective function given by

equation (II) with respect to \mathbf{w} , b (support vector) and ξ (slack variable for non-separable data). For this purpose, we used the OpenCV Machine Learning toolbox.

$$\begin{aligned} \min_{\mathbf{w}, b, \xi} \quad & \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^N \xi_i \\ & y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{aligned} \quad (1)$$

The parameter C (the penalty parameter of the error term in equation (II) and the only optional parameter set by the user for the linear kernel SVM) is iterated on a logarithmic grid and selected based on a 10-fold cross validation estimate of error rate given by the ratio of the number of misclassified samples over the number of test samples. Since we have two possible foreground masks from clustering, we train an optimal classifier for each mask. To decide the best foreground mask for representing the positive samples, x_i , $y_i = 1$ we further examine the collected positive and negative samples using the trained classifiers. The idea behind our approach is that if a mask A is representing x_i , $y_i = 1$ faithfully, then the training error of the classifier with features collected from A should be smaller than that of the classifier with features collected from the other mask, B . We measure the training error of each classifier and select the foreground mask that results in lower training error. The optimal classifier with the selected mask is used to aid further lattice detection, an advancement from [1]. Finally, we consider the foreground mask when determining image likelihood during the lattice point inference procedure, increasing accuracy in localization of lattice points. The procedure repeats until no more texels are found. Our proposed method has a 30% improved detection rate¹ over the state-of-the-art algorithm [5] on the 32 images from the PSU NRT database.

4 Multi-view and Symmetry Augmented Inpainting

One of the most challenging problems in inpainting is the scarcity of source samples [1]. We seek to overcome this in two ways. The first approach is to try to see the occluded object in another view. It is reported by Liu et al. [1] that overall occupation of the foreground fence layer in their data set is from 18% to 53%. However, even a small offset of the camera can reveal pixel values behind the foreground layer since objects behind the layer will experience less parallax than the foreground. Also, moving objects will reveal parts of themselves, even to a stationary camera, through multiple frames. Since in video these offsets are small, object alignment can be approximated as a 2D translation. We utilize the information from multiple views to aid the inpainting process by minimizing the number of pixel values that need to be inferred.

A second approach deals with the situation after multi-view inpainting or where no additional views are available. For gaps that still remain, we adopt an exemplar based inpainting algorithm [3] [4] as our base tool. In addition, we

¹ The detection rate is measured by the ratio of the number of correctly detected texels over the total number of ground truth texels.

seek to overcome scarcity of candidate patches by simulating bilateral symmetry patterns from the source image. As reflection symmetry often exists in man-made environments and nature, simulating these patterns from the source image often recovers occluded regions reliably and efficiently.

4.1 Multi-view Inpainting

To begin fence removal, we first remove the foreground layer (section 3) and then start extracting patches for inpainting. Since the order of synthesis is critical, the method for determining order that appears in [14] is used. That is, any objects that are closer or have moved more between views should be dealt with first because of their depth or motion boundary. Although optical flow estimation is often not robust due to hole regions, errors are generally not noticeable in the resulting image.

For a given image, I , we compute magnitude of optical flow, F , using the Lucas Kanade algorithm [17] for every pixel. The priority of the matching follows a descending order with respect to F . From the location p with the maximum F , we extract patch Φ_p to do block matching with the other view, I' . Formally, we seek to solve,

$$\Phi_{\hat{q}} = \arg \min_{\Phi_q \in I'} SSD(\Phi_p, \Phi_q) \quad (2)$$

where SSD is sum of squared difference.

We use a larger patch width (15~30) than the original inpainting algorithm (≤ 9) to disambiguate similar patches. This would have posed problems in earlier works [4, 13] because only complete source regions (containing no pixels to be inpainted) were considered as candidates. We allow for an area that matches better to be selected even if some of the pixels of the patch will need to be synthesized later.

Another possible problem of using a larger patch occurs at boundaries between objects at different depths. We attempt to minimize the effect of these depth boundaries by filling in the pixel values in descending order of optical flow magnitude as in [14]. Having found $\Phi_{\hat{q}}$, the value of each pixel $p \in \Phi_p \cap H$ is

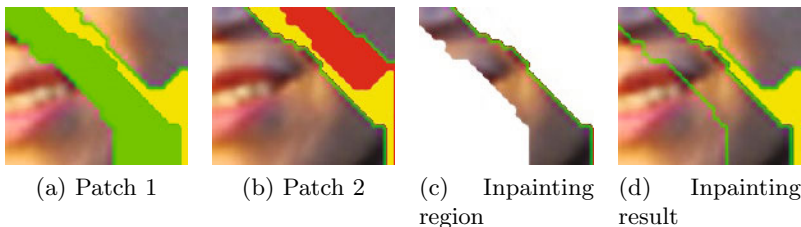


Fig. 5. Process of multiview inpainting and result: The green region shows the region that is made visible by patch 2 and the yellow region shows the region to inpaint using augmented symmetries

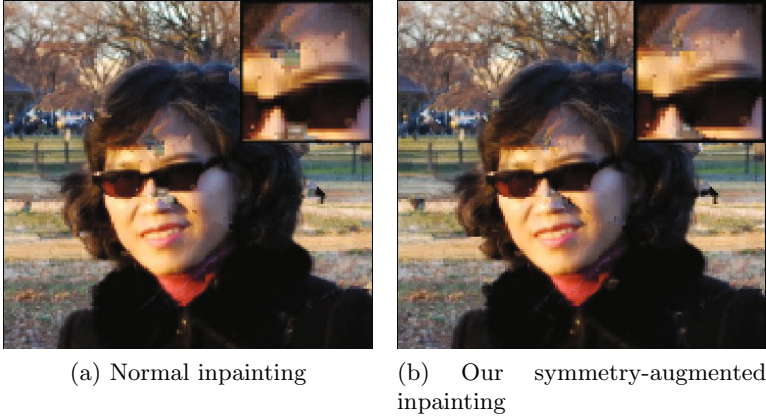


Fig. 6. Result of normal inpainting compared with symmetry-augmented inpainting. Both inpainting algorithms are applied after multi-view inpainting. (a) Results of normal inpainting [3,4] (b) Inpainting with simulated bilateral symmetry patches.

copied from its corresponding pixel $q \in \Phi_{\hat{q}} \setminus H$. If $q \in (\Phi_{\hat{q}} \setminus H)$ is the null set, the value of p is not observable from any other views, hence we use the single view inpainting algorithm in Section 4.2. As can be seen in Figure 5 we do not replace the entire original patch (Figure 5b), but only replace the region that is occluded in the original patch (Figure 5d).

4.2 Symmetry-Augmented Inpainting

After multi-view inpainting or when only one view is available, we adopt an exemplar-based single view inpainting algorithm [3,4] for hole regions that still remain. As symmetry is common in nature and man-made environments, simulating these patterns from the source image increases the pool of candidate matches, which could improve the inpainting quality. First, size of the template window Φ is given as 9 by 9 for a given image, I , and the patch priority is computed according to [3,4]. We select the patch with the highest priority, Φ_p and we rotate Φ_p by 90, 180 and 270 degrees as well as flip Φ_p around the x , y , $y = x$ and $y = -x$ axes. We next search in the source region, $S = I \setminus H$ for the patch most similar to Φ_p or its simulated symmetry patches, $\Phi_p^{(i)}$, where $i = 1 \sim 7$. Formally we seek to solve,

$$\Phi_{\hat{q}} = \arg \min_{\Phi_q \in S, i=1 \sim 7} SSD(\Phi_p^{(i)}, \Phi_q) \quad (3)$$

Having found the source exemplar $\Phi_{\hat{q}}$, we apply the appropriate inverse rotation or reflection on $\Phi_{\hat{q}}$ depending on the index, i , then the value of each pixel $p \in \Phi_p \cap H$ is copied from its corresponding location in $\Phi_{\hat{q}}$.

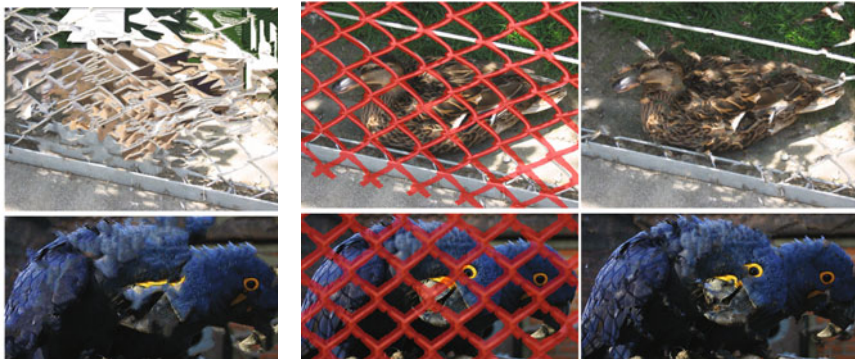
As can be seen in Figure 6, although there are still artifacts, our proposed method offers improvements in keeping the image structure (inner corner of sunglasses in Figure 6).

5 Experimental Results

We first compare our method of lattice detection to [5]. We then compare our overall system with [1] on the same images that appeared in [1]. Last, we demonstrate results of multiview and symmetry-augmented inpainting on multiview images.

Table 1. Quantitative evaluation of true positive rate and false positive rate, the true positive rate is computed by the ratio of the number of correctly identified texels over the number of ground truth texels and the false positive rate is computed by the ratio of the number of incorrectly identified texels over the number of the ground truth texels

Lattice Detection Rate	True Positive	False Positive
Park et al.	59.34% \pm 25.58	0.62% \pm 2.4
Ours	77.11% \pm 16.24	0.74% \pm 2.5



(a) Liu et al. [1]

(b) Ours

Fig. 7. Sample results of Liu et al. [1] and our approach. The middle column shows the results of our proposed segmentation method and the last column shows the results of inpainting. The results show that inpainting using a single view is still very challenging even with a good segmentation. More results can be found in “<http://vision.cse.psu.edu/research/Defencing-Revisited/index.shtml>”.

5.1 Lattice Detection

We have tested on 32 images from the PSU NRT database² [18, 19] and have found a 30% improvement in detection rate over [5]. Quantitative evaluation of true positive rate and false positive rate are shown in Table 1. The true positive rate is computed by the ratio of the number of correctly identified texels over the number of ground truth texels, and the false positive rate is computed by the ratio of the number of incorrectly identified texels over the number of ground truth texels. The ground truth data and automatic evaluation code is obtained from the PSU Near Regular Texture Database².

² <http://vision.cse.psu.edu/data/data.shtml>

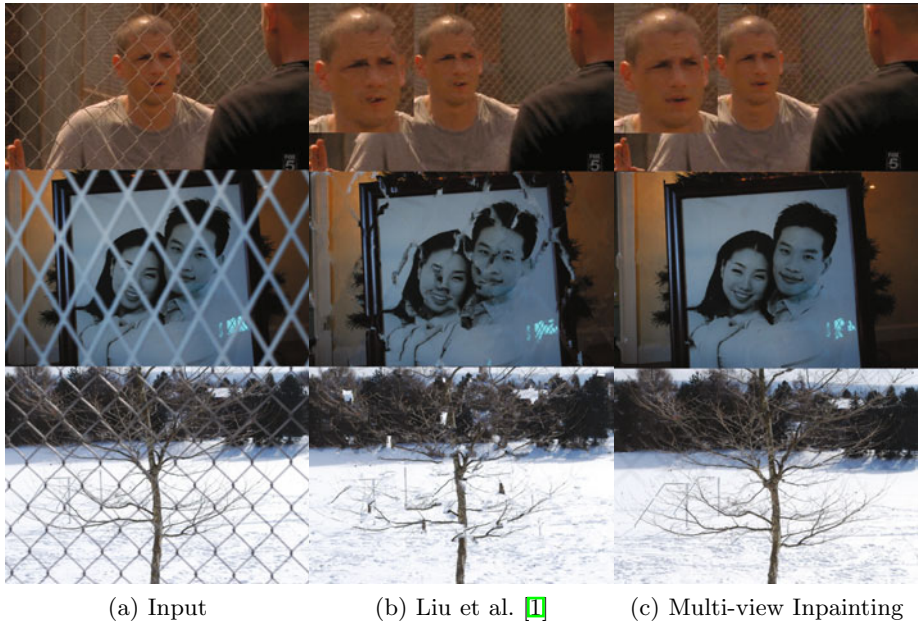


Fig. 8. (a) Input images (b) Results of [1] using a single view. (c) Results of our proposed multiview and symmetry-augmented inpainting method. The 1st row, 2nd row, and 3rd row in (c) uses 4, 3, and 2 views respectively. More results can be found in “<http://vision.cse.psu.edu/research/Defencing-Revisited/index.shtml>”.

5.2 Comparison with Liu et al. [1]

Our proposed method is successful at finding lattices and corresponding masks for both of the images that appeared in [1]. Sample results of [1] and our results³ are shown in Figure 7.

5.3 Multi-view Inpainting Result

We apply our multi-view inpainting and symmetry augmented inpainting to images that have multiple views and a few frames extracted from the show “Prison Break”. The results are illustrated in Figure 8. In Figure 8, the first row uses 4 views, the second row uses 3 views, and the last row uses 2 views.

6 Conclusion

We introduce a novel technique for “image de-fencing”, the automatic removal of foreground fence layer in real photos, by detecting, segmenting and inpainting repeating foreground structures. We treat detection and segmentation of the lattice

³ More results can be found in <http://vision.cse.psu.edu/research/Defencing-Revisited/index.shtml>

as a coupled learning process since the results of each one can be fed to the other to improve the overall performance. Our lattice detection method produces improved results over the state-of-the-algorithm [5] by 30%. We also propose multi-view inpainting and symmetry-augmented inpainting methods to overcome the problem of candidate sample patch impoverishment for inpainting. Even for human faces, these new alternatives lead to acceptable results (Figure 8). Our future goal is to deal with large view angle changes between multiple views.

Acknowledgement. This work is supported in part by an NSF grant IIS-0729363 and a gift grant to Dr. Liu from the Northrop Grumman Corporation.

References

1. Liu, Y., Belkina, T., Hays, J., Lublinerman, R.: Image de-fencing. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008, Anchorage, Alaska, pp. 1–8 (2008)
2. Hays, J., Leordeanu, M., Efros, A.A., Liu, Y.: Discovering texture regularity as a higher-order correspondence problem. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 522–535. Springer, Heidelberg (2006)
3. Criminisi, A., Perez, P., Toyama, K.: Object removal by exemplar-based inpainting. In: CVPR, pp. 721–728 (2003)
4. Criminisi, A., Perez, P., Toyama, K.: Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing* 13, 1200–1212 (2004)
5. Park, M., Collins, R.T., Liu, Y.: Deformed Lattice Discovery via Efficient Mean-Shift Belief Propagation. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 474–485. Springer, Heidelberg (2008)
6. Liu, Y., Collins, R.T., Tsin, Y.: A computational model for periodic pattern perception based on frieze and wallpaper groups. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26, 354–371 (2004)
7. Leung, T., Malik, J.: Detecting, localizing and grouping repeated scene elements from an image. In: Buxton, B.F., Cipolla, R. (eds.) ECCV 1996. LNCS, vol. 1065, pp. 546–555. Springer, Heidelberg (1996)
8. Schaffalitzky, F., Zisserman, A.: Geometric grouping of repeated elements within images. In: *Shape, Contour and Grouping in Computer Vision*, pp. 165–181 (1999)
9. Han, J., McKenna, S., Wang, R.: Regular texture analysis as statistical model selection. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 242–255. Springer, Heidelberg (2008)
10. Lin, H.C., Wang, L.L., Yang, S.N.: Extracting periodicity of a regular texture based on autocorrelation functions. *Pattern Recognition Letters*, 433–443 (1997)
11. Leonard, G.O.H., Takeo, K.: Computer analysis of regular repetitive textures. In: *Proceedings of a Workshop on Image Understanding Workshop*. Morgan Kaufmann Publishers Inc., San Francisco (1989)
12. Favaro, P., Grisan, E.: Defocus inpainting. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 349–359. Springer, Heidelberg (2006)
13. Wexler, Y., Shechtman, E., Irani, M.: Space-time completion of video. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29, 463–476 (2007)
14. Patwardhan, K.A., Sapiro, G., Bertalmio, M.: Video inpainting of occluding and occluded objects. In: *IEEE International Conference on Image Processing, ICIP 2005*, vol. 2, pp. 69–II-72 (2005)

15. James, H., Alexei, A.E.: Scene completion using millions of photographs, 12763824 (2007)
16. Vaish, V., Levoy, M., Szeliski, R., Zitnick, C.L., Sing Bing, K.: Reconstructing occluded surfaces using synthetic apertures: Stereo, focus and robust measures. In: *Computer Vision and Pattern Recognition*, vol. 2, pp. 2331–2338 (2006)
17. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: *Proc. of 7th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 674–679 (1981)
18. Park, M., Lee, S., Chen, P.C., Kashyap, S., Butt, A.A., Liu, Y.: Performance evaluation of state-of-the-art discrete symmetry detection algorithms. In: *Proceedings of CVPR 2008* (2008)
19. Chen, P.C., Hays, J.H., Lee, S., Park, M., Liu, Y.: A quantitative evaluation of symmetry detection algorithms. Technical Report CMU-RI-TR-07-36, Robotics Institute, Pittsburgh, PA (2007)

Feature-Assisted Dense Spatio-temporal Reconstruction from Binocular Sequences

Yihao Zhou and Yan Qiu Chen

School of Computer Science
Fudan University, Shanghai, China
{yihzhou, chenyaq}@fudan.edu.cn

Abstract. In this paper, a dynamic surface is represented by a triangle mesh with dense vertices whose 3D positions change over time. These time-varying positions are reconstructed by finding their corresponding projections in the images captured by two calibrated and synchronized video cameras. To achieve accurate dense correspondences across views and frames, we first match sparse feature points and rely on them to provide good initialization and strong constraints in optimizing dense correspondence. Spatio-temporal consistency is utilized in matching both features and image points. Three synergistic constraints, image similarity, epipolar geometry and motion clue, are jointly used to optimize stereo and temporal correspondences simultaneously. Tracking failure due to self-occlusion or large appearance change are automatically handled. Experimental results show that complex shape and motion of dynamic surfaces like fabrics and skin can be successfully reconstructed with the proposed method.

1 Introduction

Dynamic surfaces undergoing complex motion prevalently exist in nature. Typical examples include fluttering fabrics and deforming skin. A deforming surface can be represented by a spatio-temporal model, i.e. a polyhedral mesh whose vertex positions vary over time. At a given time step, the mesh represents the instantaneous 3D shape of the surface, and the time-varying 3D positions of each vertex represent its motion trajectory in the entire time span. Acquiring accurate dense spatio-temporal model of a deforming surface is highly useful in many applications such as realistic computer animation [18], study of emotional facial expressions [15], and investigation of mechanical properties of materials [5].

The most popular approach to dense spatio-temporal reconstruction is to use multiple synchronized video cameras due to its non-contact characteristic and sufficient spatial and temporal resolution. Most existing approaches [1,2,8,13,14] start from reconstructing initial 3D shape by multiview silhouettes or multiview stereo, and then estimate the corresponding positions of the reconstructed points in subsequent frames. This strategy has several drawbacks. First, although multiview silhouettes can be used to recover approximate structure of objects like human head and body [1,2,13], it is not applicable to obtaining fine structure of surfaces such as fabric and facial skin. Second, multiview stereo algorithms [8,14]

could suffer severe stereo matching ambiguities in binocular setup, leading to very noisy reconstructed 3D shapes. Third, dense motion trajectories are usually computed based on previously estimated 3D shapes [1, 2, 8, 14]. Therefore 3D reconstruction error is inherited in motion estimation and the synergistic relationship between shape and motion is neglected.

In this paper, we represent a dynamic surface as a triangle mesh with dense vertices whose positions change with time. The 3D motion trajectories of the vertices are reconstructed by finding their corresponding projections in the images recorded by a pair of calibrated and synchronized video cameras. Spatial correspondences (across viewpoints) are triangulated to compute their instantaneous 3D positions, and temporal correspondences (across frames) construct their motion trajectories. In order to achieve accurate dense correspondences in images of different views and frames, we rely on matched image feature points to guide dense correspondence computation. The main advantages of this feature-assisted framework are that (1) feature points can be robustly matched despite large variation of image appearance due to change of viewpoint and complex surface deformation, (2) matched feature points can give strong constraint for correspondences of image points in the vicinity, and (3) they provide fairly good initialization for the optimization of dense correspondences, making it much less susceptible to local minima. In addition, in order to utilize the synergistic relationship between shape and motion constraints, we use spatio-temporal consistency to jointly optimize spatial and temporal correspondences. Three constraints, image similarity, epipolar geometry and motion clue, are incorporated into a single cost function which is minimized to obtain optimal correspondences of both sparse feature points and dense image points. Tracking lost due to self-occlusion or large variation in image appearance and reoccurrence of lost points are automatically detected.

The main contributions of our work include:

- A feature-assisted framework is developed for reconstructing dynamic surfaces, in which sparse matched features efficiently guide dense spatio-temporal reconstruction.
- When matching both sparse features and dense image points, the synergistic relationship between shape and motion constraints is utilized to disambiguate stereo matching and temporal tracking simultaneously.

2 Related Work

Apart from the aforementioned methods, several methods are related to the proposed approach as well. In [9, 11, 17], stereo disparity and optical flow are coupled to compute dense stereo and temporal correspondences in binocular image sequences. In [16], optical flows individually measured in numerous views are matched to compute 3D scene flow between adjacent frames. Unfortunately, only instantaneous 3D motion field between two frames is addressed, instead of spatio-temporal reconstruction which involves recovering the vertex motion trajectories across the entire time span.

In [4], the dynamic scene is represented by a collection of surfels, each of which encodes 25 parameters modeling its shape, reflectance and motion. However, in order to recover these parameters, the surfel must correspond to a large set of image pixels, and therefore only a sparse reconstruction of the scene is obtained. In [6], corner points detected in binocular sequences are matched by jointly using epipolar geometry and motion constraint, and their corresponding 3D motion trajectories are reconstructed. Similar to [4], the resulting point set is too sparse to represent the complex structure and deformation of the dynamic surface.

Similar to our approach, several methods [1,2] use known correspondences to initiate dense motion field between adjacent frames. However, instead of being accurately estimated, dense motion field is interpolated from sparse 3D correspondences. In contrast, our approach use matched feature points as constraints for dense correspondences instead of direct interpolation. While matching both sparse feature points and dense image points, three synergistic constraints, image similarity, epipolar constraint and motion clue, are jointly used, so that consistent spatial and temporal correspondence are simultaneously optimized. Experimental results show that the proposed method can successfully recover both complex 3D shape and highly non-rigid motion of deformable surfaces such as fabric and skin.

3 Method

3.1 The Spatio-temporal Reconstruction Problem

Consider a surface undergoing complex non-rigid motion. We represent its deformation during time span $[1, T]$ by time-varying 3D coordinates of a dense set of N surface sample points $\{X_t^i, i = 1, 2, \dots, N, t = 1, 2, \dots, T$. The dynamic surface is recorded by a pair of calibrated and synchronized video cameras, producing two image sequences $\{I_t^l\}$ and $\{I_t^r\}$, where the superscripts denote left and right view respectively. In this case, the 3D motion trajectories of the surface points are reconstructed by finding their corresponding image projections. Since the 3D surface shape is unknown at first, we specify a region in image I_1^l of the first frame to indicate the surface part to be reconstructed, and a dense set of image points $\{x_1^l\}$ are uniformly sampled in this region. Our goal is to find their corresponding image locations in all the other captured images.

3.2 Feature-Assisted Framework

We develop a feature-assisted framework to establish dense correspondence in a pair of adjacent frames. The feature points in the underlying four images are first matched and then they are used to guide dense correspondences. More concretely, sparse matched features give fairly good initial estimates of dense correspondences, and introduce strong constraints for the optimization of them. In both matching feature points and dense image points, spatio-temporal consistency, which incorporates image similarity, epipolar geometry and motion clue,

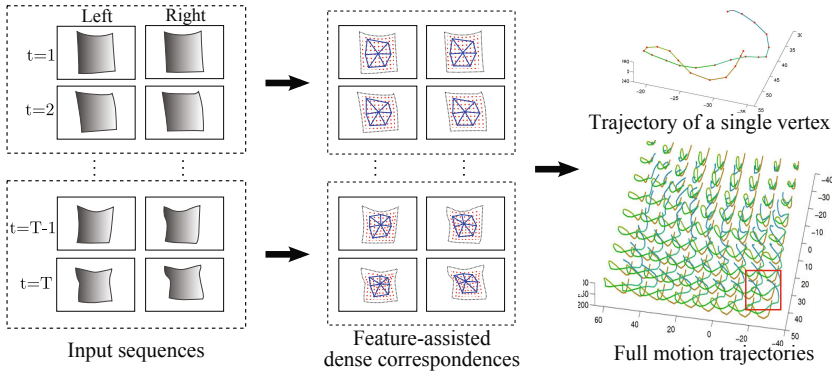


Fig. 1. Feature assisted dense spatio-temporal reconstruction. Blue connected points represent sparse matched feature points which provide strong evidence of matching dense image points (in red) in their vicinity. The dotted square denote that the spatial and temporal correspondences in adjacent frames are jointly optimized using spatio-temporal consistency. The full motion trajectories (from brown to blue) are reconstructed frame by frame.

is exploited to simultaneously optimize stereo matching and temporal correspondences (which will be described in the following section). The resulting stereo correspondences are used for 3D reconstruction and the whole set of 3D motion trajectories are obtained by performing the feature-assisted algorithm frame by frame (see Fig. 1).

3.3 Spatio-temporal Consistency

We use spatio-temporal consistency to represent a set of constraints for corresponding points both spatially matched (across views) and temporally matched (across frames). These constraints include image similarity, epipolar geometry, and motion clue. Concretely, in two consecutive frames, if $(x_t^l, x_t^r, x_{t+1}^l, x_{t+1}^r)$ are image locations corresponding to a same surface point, they must satisfy the following conditions (see Fig. 2):

- *Image similarity*: Assuming the surface is Lambertian, the image patches must be highly correlated around each spatial pair (x_t^l, x_t^r) and (x_{t+1}^l, x_{t+1}^r) as well as each temporal pair (x_t^l, x_{t+1}^l) and (x_t^r, x_{t+1}^r) .
- *Epipolar geometry*: Each spatial pair (x_t^l, x_t^r) and (x_{t+1}^l, x_{t+1}^r) must satisfy the epipolar constraint.
- *Motion clue*: We assume that surface points in a small neighborhood undergo similar (not necessarily same) motion. This constraint is satisfied for a variety of deformable surfaces like fabric and skin. It is more generalized than the piecewise rigidity model [8], and the entire surface can still deform in a highly non-rigid manner.

Although the above constraints seem disparate, jointly using them can use the synergistic relationship between them to largely mitigate the ambiguities in finding spatio-temporal consistent correspondences. Thus, we design a cost function

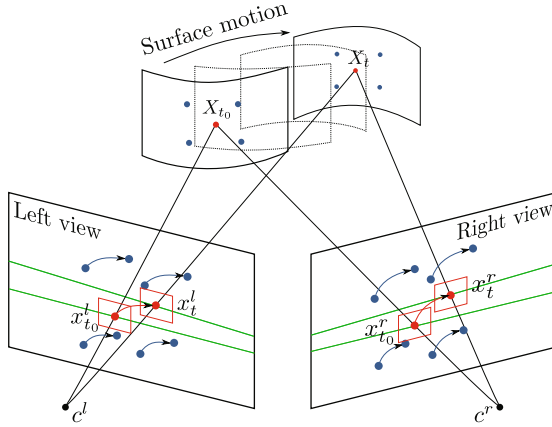


Fig. 2. Spatio-temporal consistency. Red squares indicate image similarity constraint. Green lines represent the epipolar lines. Blue points in the vicinity undergo similar motion as the underlying red points.

for a 4-tuple of four image positions $q = (x_t^l, x_t^r, x_{t+1}^l, x_{t+1}^r)$

$$E(q) = \alpha E_{img}(q) + \beta E_{epi}(q) + \gamma E_{mot}(q) \tag{1}$$

where the three components are associated with the above constraints respectively. The optimal 4-tuple can be derived by minimizing the cost function. The details will be given in the following sections.

3.4 Matching Feature Points Using Spatio-temporal Consistency

In the four images of two consecutive frames, we first extract SIFT features [12] as their descriptors provide robust matching despite variation in image appearance due to view changing and surface deformation. For each feature point x_t^l in image I_t^l , we wish to find its correspondences in the other three feature sets, such that the 4-tuple $q = (x_t^l, x_t^r, x_{t+1}^l, x_{t+1}^r)$ minimizes the cost function (1). The detailed formulation is presented as follows.

Image Similarity. To define the term $E_{img}(q)$, we utilize the discriminative SIFT descriptors. Smaller descriptor distance suggests higher similarity of the image patches around the features. We therefore formulate the term $E_{img}(q)$ as

$$E_{img}(q) = \frac{1}{4} \left[\rho_d(x_t^l, x_t^r) + \rho_d(x_{t+1}^l, x_{t+1}^r) + \rho_d(x_t^l, x_{t+1}^l) + \rho_d(x_t^r, x_{t+1}^r) \right] \tag{2}$$

where $\rho_d(\cdot)$ represents the Euclidean distance between two feature descriptors.

Epipolar Geometry. As each spatial pair must satisfy epipolar constraint, the term $E_{epi}(q)$ is formulated as

$$E_{epi}(q) = \frac{1}{2} \left[\psi_e(\rho_e(x_t^l, x_t^r)) + \psi_e(\rho_e(x_{t+1}^l, x_{t+1}^r)) \right] \tag{3}$$

where $\rho_e(\cdot)$ represents the average distance of the two points to their respective epipolar lines, and $\psi_e(\cdot)$ is a sigmoid-like function for thresholding

$$\psi_e(x) = 1 - \frac{2}{1 + e^{\lambda(x - \sigma_e)}} \tag{4}$$

where λ is a positive value ($\lambda = 3$ in our implementation) and σ_e is the threshold.

Motion Clue. Although we constrain in Section 3.3 that nearby points undergo similar motion, this clue can not be fully utilized so far, since no motion has been recovered yet in the vicinity of a feature point. Instead, we assume that the temporal correspondence of x_t^l (or x_t^r) is within a distance σ_m . The motion clue term is therefore written as

$$E_{mot}(q) = \frac{1}{2} \left[\psi_m(\|x_{t+1}^l - x_t^l\|) + \psi_m(\|x_{t+1}^r - x_t^r\|) \right] \tag{5}$$

where $\psi_m(\cdot)$ is a function similar to Eq. (4) with threshold σ_m .

Filtering. As the correspondences are individually found for each feature $x_t^{l,i}$, we filter out outliers using an additional smoothness constraint. Ideally, the four features in a 4-tuple q^i correspond to a same surface point with different 3D positions at t and $t + 1$. Thus, the acquired set of 4-tuples $\{q_t^i\}$ correspond to two 3D point sets, which are sparse representation of the instantaneous 3D surface shapes at t and $t + 1$ respectively. We compute the two 3D point sets $\{X_t^i\}$ and $\{X_{t+1}^i\}$ using triangulation and discard outliers by enforcing piecewise smoothness on each instantaneous shape. We discard any point X_t^i if

$$\frac{|d_t^i - \tilde{d}|}{\text{med}_{j \in \Omega_p(X_t^i)} |d_t^j - \tilde{d}|} > \epsilon_p \tag{6}$$

where d_t^i is the depth value of X_t^i , $\Omega_p(X_t^i)$ is the set of k_p points closest to X_t^i , and \tilde{d} is the median depth of these points.

So far, we have acquired a set of matched features $\{(a_t^{l,i}, a_t^{r,i}, a_{t+1}^{l,i}, a_{t+1}^{r,i})\}$ with known correspondences in frame t and $t + 1$, as well as their corresponding 3D positions $\{(A_t^i, A_{t+1}^i)\}$. They act as anchor points in the next stage where the correspondence for an arbitrary image location is to be computed. The benefits of using anchoring feature points include: (1) they provide good initials for the optimization, making it much less susceptible to local minima, and (2) they provide strong constraint for defining the cost function of image points in the vicinity.

3.5 Feature-Assisted Dense Correspondences

After acquiring the anchor points, we can exploit them to guide the optimization of dense correspondences. For an arbitrary image point x_t^l , we compute the optimal 4-tuple $q = (x_t^l, x_t^r, x_{t+1}^l, x_{t+1}^r)$ again by minimizing the cost function (1). However, we modify the formulation of $E_{img}(q)$ and $E_{mot}(q)$ used in Section 3.4 in order to make use of the matched feature points.

Image Similarity. While feature descriptor distance is used in the previous section, computing descriptor for each image location is impractical and unnecessary. Instead, we use windowed normalized cross correlation as a measure of the similarity of two image patches. The term $E_{img}(q)$ is formulated as

$$E_{img}(q) = \frac{1}{4} \left[\Delta(x_t^l, x_t^r) + \Delta(x_{t+1}^l, x_{t+1}^r) + \Delta(x_t^l, x_{t+1}^l) + \Delta(x_t^r, x_{t+1}^r) \right] \quad (7)$$

$$\Delta(x_1, x_2) = \frac{-\sum_{dx} [I_1(x_1 + dx) - \bar{I}_1] [I_2(x_2 + H(dx)) - \bar{I}_2]}{\sqrt{\sum_{dx} [I_1(x_1 + dx) - \bar{I}_1]^2} \sqrt{\sum_{dx} [I_2(x_2 + H(dx)) - \bar{I}_2]^2}} \quad (8)$$

where dx is the image location in a squared window of size $R \times R$ centered at $(0, 0)$, and $H(\cdot)$ is an affine transform estimated by neighboring 2D anchor points. The minus sign in the numerator is used because higher image similarity should give lower cost.

Motion Clue. Given a sparse set of anchor points, the motion clue in Section 3.3 can be used. The set of 3D anchors $\{(A_t^i, A_{t+1}^i)\}$ can be regarded as a sparse representation of the dynamic surface in frame t and $t + 1$. For a 3D point X_t inside the triangle (A_t^i, A_t^j, A_t^k) with barycentric coordinates $\mathbf{b} = [b_1 \ b_2 \ b_3]^T$, $\hat{X}_{t+1} = b_1 A_{t+1}^i + b_2 A_{t+1}^j + b_3 A_{t+1}^k$ is a good estimate for its correspondence at frame $t + 1$. Similarly in 2D domain, for an image point x_t^l in the triangle $(a_t^{l,i}, a_t^{l,j}, a_t^{l,k})$, the 2D positions $\hat{x}_t^r, \hat{x}_{t+1}^l, \hat{x}_{t+1}^r$, having barycentric coordinates \mathbf{b} w.r.t. their respective enclosing anchors, are fairly good estimates for the correspondences of x_t^l . As such, we constrain the distance between optimal correspondence to its initial estimate is smaller than a threshold σ_b . The motion clue term is therefore written as

$$E_{mot}(q) = \frac{1}{3} \left[\psi_b(\|x_t^r - \hat{x}_t^r\|) + \psi_b(\|x_{t+1}^l - \hat{x}_{t+1}^l\|) + \psi_b(\|x_{t+1}^r - \hat{x}_{t+1}^r\|) \right] \quad (9)$$

where $\psi_b(\cdot)$ is sigmoid-like function with threshold σ_b .

3.6 Handling Tracking Failure and Reoccurrence of Lost Points

Although anchor points greatly facilitate dense matching, falsely matched features or insufficient anchors in the vicinity can produce inferior results for an image point. We therefore examine the resulting image correlation E_{img} for each image point x_t^l . If $E_{img} > \epsilon_E$, we recompute the affine transform and initial correspondences $\hat{x}_t^r, \hat{x}_{t+1}^l, \hat{x}_{t+1}^r$ by using successfully matched image points in the neighborhood, instead of the potentially incorrect anchoring features.

Nonetheless, tracking failure may still be caused by self-occlusion, depth discontinuity, or large appearance change. To solve this problem, we select a reference frame t_r (the first frame in our experiments) in which the surface region to be tracked are assumed to be visible in both views. At frame t , we label any surface point as *tracking failure* if its image correlation $E_{img} > \epsilon_E$. In case this lost point reoccurs in the next frame $t + 1$, we attempt to find in frame $t + 1$

the location corresponding to its position in the reference frame t_r . To this end, the same optimization is performed except that the neighboring image points labeled as *tracking success* are used as anchors (note that in general, no matched features are available for frame t_r and $t + 1$). If the resulting $E_{img} \leq \epsilon_E$, the surface point is labeled as *tracking success* in frame $t + 1$.

4 Implementation Details

In section 3.4, a cost function is minimized to find optimal correspondences for a feature point. However, exhaustive search in all possible 4-tuple is computationally expensive due to the large number of features (typically 2000-7000). Instead, we use a greedy strategy, utilizing the discriminative feature descriptors. When searching for the spatial or temporal correspondence for a feature point, we only regard the k_f (3-5 in our experiments) features nearest to the best matched feature as matching candidates. The best spatial match (across views) is the one with minimal descriptor distance out of all the features whose deviation from the epipolar line is less than ϵ_e . The best temporal match (across frames) is the one with minimal descriptor distance out of all the features within distance σ_m (see Fig. 3). After the correspondences are computed for each feature point x_t^l , false matches are filtered out (Section 3.4). The remaining unmatched feature points are plugged into the next iteration. In our experiment, the whole procedures are iterated for five times.

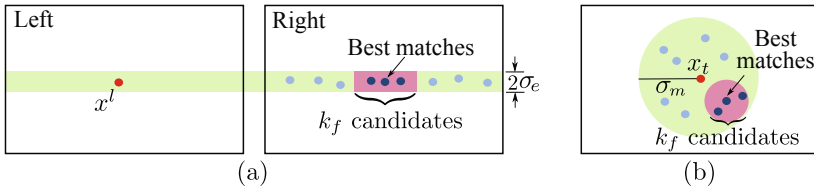


Fig. 3. Strategy for matching feature points. (a) Candidates for spatial correspondence of x^l . (b) Candidates for temporal correspondence of x_t .

When searching for correspondences of image points, we first process the interior points inside the 2D anchor mesh constructed by delaunay triangulation of $\{a_t^l\}$. Simplex search method [10] is used for optimization described in Section 3.5. The points with resulting $E_{img} < \sigma_E$ are deemed as tracking success. For the remaining untracked points, neighboring successfully matched points are used as anchors instead of feature points. Same optimization is performed and those with $E_{img} < \sigma_E$ are labeled as tracking success. Finally, the reference frame t_r is used to detect whether any lost point reoccurs in frame $t + 1$.

5 Experimental Results

We test the proposed method using several challenging datasets and compare with existing methods. Both synthetic and real-world cases are used for evaluation. The parameters used in the experiments are given in Table 1.

Table 1. The parameters used in the experiments

Dataset	α	β	γ	k_f	σ_e	σ_m	k_p	ϵ_p	R	σ_b	ϵ_E
cloth	0.3	0.3	0.3	5	3	50	10	3	15	5	0.7
flag	0.3	0.3	0.3	5	3	50	5	10	21	5	0.7
skin	0.3	0.3	0.3	5	3	50	5	3	21	5	0.7

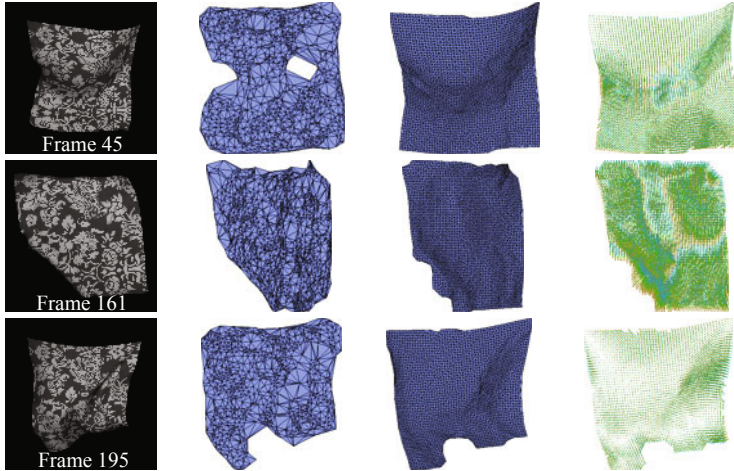


Fig. 4. Reconstruction results in three frames where severe self-occlusion or large deformation occur. From left to right: input images, sparse matched feature points, dense reconstructed points, and instantaneous 3D motion fields (from brown to blue).

5.1 Simulated Cloth Fluttering in the Wind

In order to give quantitative evaluation of our method, we use *3DS MAX* to simulate a textured piece of cloth of size 100cm×100cm waving in the wind. Two virtual cameras are placed to capture image sequences of 200 frames at 800×600 pixel resolution. The speed of the wind is set to be varying so that both instantaneous shapes and motion are highly complex. In addition, self-occlusion occurs frequently, which makes the spatio-temporal reconstruction even more challenging. The results of several frames are shown in Fig. 4. Although some surface parts are lost due to self-occlusion, large image variation or shadows, the time-varying positions of a very dense set of sample points are recovered.

In order to show the advantage of the proposed method, we compare the correspondences (1) calculated by state-of-the-art dense optical flows [3], (2) initially estimated by matched feature points, and (3) optimized by feature-assisted spatio-temporal consistency. The positions of the $19 \times 19 = 361$ control points on the surface are used as ground truth. For optical flows, the ground truth of image projections in the first frame are given, and their correspondences in any subsequent frame are obtained using 2D flows individually estimated in each view. Correspondences in the same frame are triangulated to derive

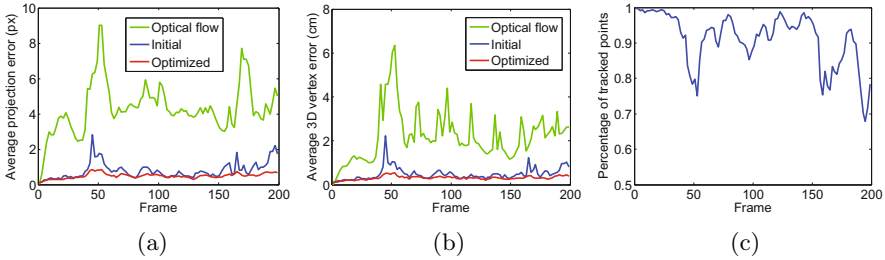


Fig. 5. Reconstruction accuracy and completeness. (a) Average pixel distance between image projections and ground truth. (b) Average distance between 3D vertices and ground truth. (c) Percentage of successfully tracked points.

instantaneous 3D positions. To measure the reconstruction accuracy, the average distance of successfully tracked points to the ground truth is used. Fig. 5 shows that the reconstruction error of dense flows is significantly higher than the other two methods. We also notice that in most frames, the correspondences initialized by anchor points are comparable to the optimized ones, which demonstrates that matched feature points can provide fairly good initial estimates of dense correspondences. However, in frames where severe self-occlusion occurs, initial estimates give noisy results. The feature-assisted spatio-temporal reconstruction yields best accuracy. The average projection distance is below 1 pixel, and the 3D reconstruction error remains at a significantly low level ($<0.5\text{cm}$, 0.35% of the cloth diagonal). In addition, no significant error drift is shown and the percentage of successfully tracked points drops as expected due to large deformation and self-occlusion (see Fig. 4).

5.2 Flag Sequence

The first real-world scene used to assess the quality of our method is the *flag* dataset from [4]. Sequences of view 1 and 2 are chosen as the input and each stream consists of 37 images of resolution 722×482 . Dense spatio-temporal reconstruction from this binocular sequences is highly challenging due to (1) frequent self-occlusion, (2) uniform or slowly-varying colors in many regions, (3) fast and highly complex motion, and (4) very few viewpoints available.

The results are shown in Fig. 6. Both instantaneous 3D shapes and full 3D motion trajectories are successfully recovered although only two views are used. Note that one of the best multi-view stereo algorithm [7] produces fairly noisy 3D shapes in this binocular setting. Consequently, motion recovery based on this initial shape [8] is expected to be inaccurate. In contrast, our method regularize dense correspondences by utilizing feature points which can be robustly matched despite large image variation due to view changes and surface deformation. Spatio-temporal consistency is also used to disambiguate both stereo matching and temporal tracking.

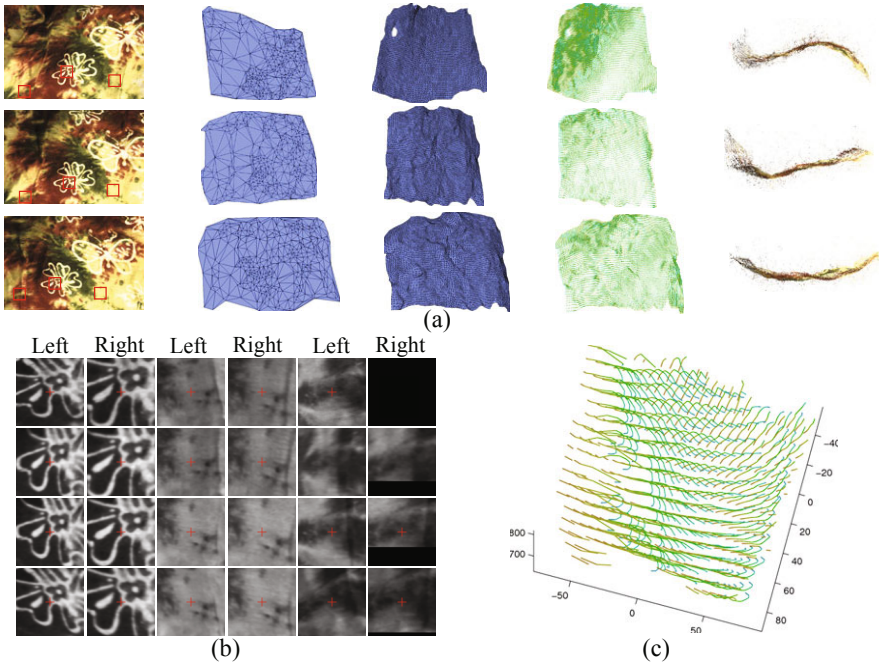


Fig. 6. Result of the flag sequences. (a) Results of frame 5,10 and 15. From left to right: input images, sparse matched feature points, dense reconstructed points, instantaneous 3D motion fields, and 3D reconstruction by PMVS [7]. (b) Image projections of three sample points denoted by red squares in the input images. (c) 3D motion trajectories (only a portion of them are plotted for better visualization).

In [4], the reconstructed surface consists of only about 200 dynamic surfels, because each surfel must correspond to a large set of pixels so that the 25 parameters encoded in each surfel can be reliably recovered. In contrast, our method generates the 3D motion trajectories of a very dense of surface points.

Since no ground truth data is available, we select three sample points and visualize their image projections in both views to illustrate the reconstruct accuracy. These sample points lie in three representative regions: (1) sufficiently textured, (2) poorly textured, and (3) lost in several frames. We see that accurate correspondences are obtained in all these surface parts.

5.3 Facial Skin Deformation

We also test our method on a more challenging real-world case: facial skin deformation around a human mouth. Unlike fabrics, facial skin is frequently stretched during its deformation, leading to large variation in image appearance. Due to lack of sufficient texture, black letters are randomly painted on the skin. The capture system is composed of two synchronized SONY HVR-V1C video

cameras delivering 25 fps at 1440×1080 pixel resolution. We choose 26 frames during which the mouth is stretched toward both sides and then moves forward. The images are resized to 720×540 pixel resolution and we use gray-level images for all computations. The two cameras are geometrically calibrated using the method proposed in [19]. The surface area to be tracked is manually specified and the regions corresponding to noise, lips and mouth are discarded. Fig. 7 shows that the 3D motion trajectories of dense surface points are successfully recovered. The projections of three sample points are visualized. One of them is lost in several frames since it moves outside the field of view. The other two lie in regions whose appearance change significantly during deformation.

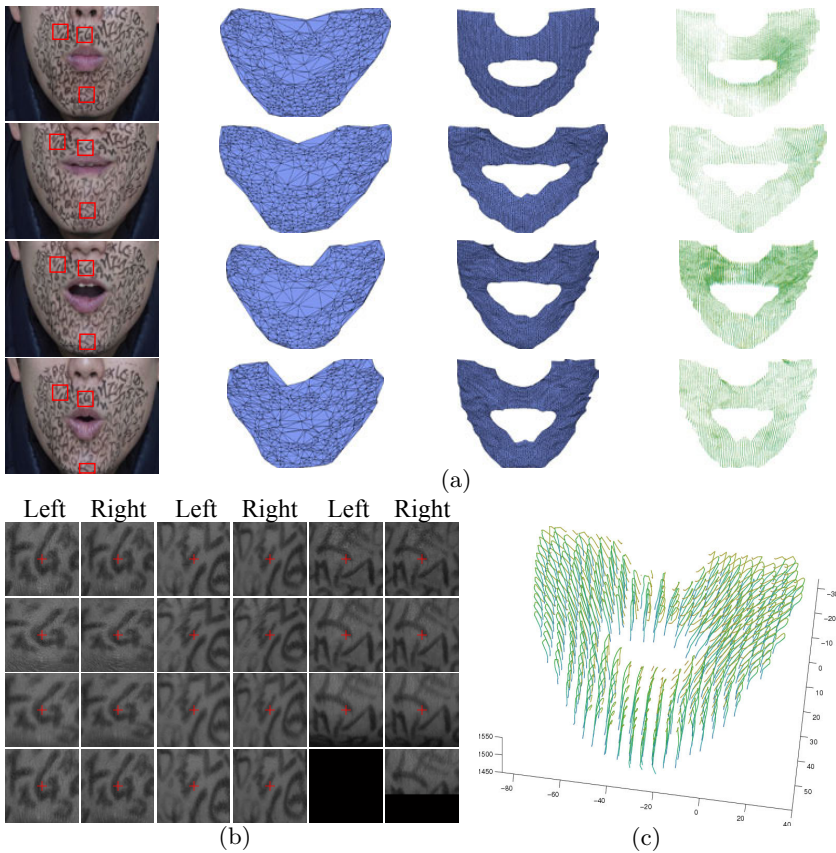


Fig. 7. Result of skin deformation. (a) Results of frame 1,5,10 and 15. From left to right: input images, sparse matched feature points, dense reconstructed points, and instantaneous 3D motion fields. (b) Image projections of three sample points denoted by red squares in the input images. (c) 3D motion trajectories (only a portion of them are plotted for better visualization).

6 Conclusion

In this paper, we recover dense spatio-temporal model by finding dense correspondences in the images captured by a binocular stereo system. A feature-assisted framework is developed, in which matched feature points efficiently guide dense correspondence. The advantages are remarkable in several aspects: (1) feature points are robustly matched in spite of large variation in image appearance due to different viewpoints and complex surface deformation, and (2) matched feature points provide fairly good initialization as well as strong constraints for computation of dense correspondences in their vicinity. Spatio-temporal consistency incorporates synergistic constraints including image similarity, epipolar geometry and motion clue, and it can be used to largely disambiguate stereo and temporal correspondences. Experimental results show that feature-assisted spatio-temporal reconstruction exhibits significant advantage over existing methods, and it is capable of recovering both complex structure and motion of dynamic surfaces like fabrics and skin.

Acknowledgement. The research work presented in this paper is supported by National Natural Science Foundation of China, Grant No. 60875024 Education Commission of Shanghai Municipality Grant No. 10ZZ03, and Science and Technology Commission of Shanghai Municipality, Grant No. 09JC1401500.

References

1. de Aguiar, E., Theobalt, C., Stoll, C., Seidel, H.P.: Marker-less deformable mesh tracking for human shape and motion capture. In: IEEE Conference on Computer Vision and Pattern Recognition (2007)
2. Ahmed, N., Theobalt, C., Rossl, C., Thrun, S., Seidel, H.P.: Dense correspondence finding for parametrization-free animation reconstruction from video. In: IEEE Conference on Computer Vision and Pattern Recognition (2008)
3. Brox, T., Bruhn, A., Papenber, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3024, pp. 25–36. Springer, Heidelberg (2004)
4. Carceroni, R.L., Kutulakos, K.N.: Multi-view scene capture by surfel sampling: from video streams to non-rigid 3D motion, shape and reflectance. *The International Journal of Computer Vision* 49, 175–214 (2002)
5. Chivers, K., Clocksin, W.: Inspection of surface strain in materials using optical flow. In: British Machine Vision Conference, pp. 392–401 (2000)
6. Du, H., Zou, D., Chen, Y.Q.: Relative epipolar motion of tracked features for correspondence in binocular stereo. In: IEEE International Conference on Computer Vision (2007)
7. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multi-view stereopsis. In: IEEE Conference on Computer Vision and Pattern Recognition (2007)
8. Furukawa, Y., Ponce, J.: Dense 3D motion capture from synchronized video streams. In: IEEE Conference on Computer Vision and Pattern Recognition (2008)
9. Huguet, F., Devernay, F.: A variational method for scene flow estimation from stereo sequences. In: IEEE International Conference on Computer Vision (2007)

10. Lagarias, J.C., Reeds, J.A., Wright, M.H., Wright, P.E.: Convergence properties of the Nelder-Mead simplex method in low dimensions. *SIAM Journal of Optimization* 9, 112–147 (1998)
11. Li, R., Sclaroff, S.: Multi-scale 3D scene flow from binocular stereo sequences. *Computer Vision and Image Understanding* 110, 75–90 (2008)
12. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *The International Journal of Computer Vision* 60, 91–110 (2004)
13. Neumann, J., Aloimonos, Y.: Spatio-temporal stereo using multi-resolution subdivision surfaces. *The International Journal of Computer Vision* 47, 181–193 (2002)
14. Pons, J.P., Keriven, R., Faugeras, O.: Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *The International Journal of Computer Vision* 72, 179–193 (2007)
15. Susskind, J.M., Lee, D.H., Cusi, A., Feiman, R., Grabski, W., Anderson, A.K.: Expressing fear enhances sensory acquisition. *Nature Neuroscience* 11, 843–850 (2008)
16. Vedula, S., Baker, S., Rander, P., Collins, R., Kanade, T.: Three-dimensional scene flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 475–480 (2005)
17. Wedel, A., Rabe, C., Vaudrey, T., Brox, T., Franke, U., Cremers, D.: Efficient dense scene flow from sparse or dense stereo data. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part I. LNCS*, vol. 5302, pp. 739–751. Springer, Heidelberg (2008)
18. White, R., Crane, K., Forsyth, D.A.: Capturing and animating occluded cloth. In: *SIGGRAPH* (2007)
19. Zhang, Z.: A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 1330–1334 (2000)

Improved Spatial Pyramid Matching for Image Classification

Mohammad Shahiduzzaman, Dengsheng Zhang, and Guojun Lu

Gippsland School of IT, Monash University, Australia
{Shahid.Zaman,Dengsheng.Zhang,Guojun.Lu}@monash.edu

Abstract. Spatial analysis of salient feature points has been shown to be promising in image analysis and classification. In the past, spatial pyramid matching makes use of both of salient feature points and spatial multiresolution blocks to match between images. However, it is shown that different images or blocks can still have similar features using spatial pyramid matching. The analysis and matching will be more accurate in scale space. In this paper, we propose to do spatial pyramid matching in scale space. Specifically, pyramid match histograms are computed in multiple scales to refine the kernel for support vector machine classification. We show that the combination of salient point features, scale space and spatial pyramid matching improves the original spatial pyramid matching significantly.

1 Introduction

Image classification has attracted large amount of research interest in the past few decades due to the ever increasing digital image data generated around the world. Traditionally, images are represented and retrieved using low level features. Recently, machine learning tools have been widely used to classify images into semantic categories. Now low level features can be used more efficiently than ever. Image classification is an important application in computer vision. Our research goal is to improve methods for Image classification, more specifically natural scene images or images with some spatial configurations. We want to classify an image based on its semantic category of a scene like forest, road or building etc. Our approach to whole image categorization employs to renowned techniques namely Spatial Pyramid Matching (SPM) [1] and scale space theory. Our objective is to combine the power of these two methods.

In this paper, scene categorization is attempted by global image representation developed from low level image properties. There is another approach for this task that is to get idea of high level semantic attributes by segmentation of objects on the scene (like bed or car) and classify the scene accordingly. We believe scene classification can be done without extracting this high level object cues. This is inspired by the publications of [2] where they proved that people can recognize natural scenes while overlooking most of the details in it (i.e. the constituent objects). In another publication [3] it is also shown that global information is as important as local information for scene classification by human subjects.

Scale is an important aspect of local feature finding in prominent cue detection in images. The most prominent example of using scale space and characteristics scale is the local invariant feature detector SIFT [4]. In SIFT the authors used

maxima/minima of neighboring scale space to find the interest points or key points of an image. Scene features like sands in a beach or certain textures in the curtain of a room would be more evident in bigger scales. Scale-space theory is a framework for multi-scale signal representation. It is a formal theory for handling image structures at different scales, by representing an image as a one-parameter family of smoothed images, the scale-space representation, parameterized by the size of the smoothing kernel used for suppressing fine-scale structures [5].

In recent years the bag-of-features (BoF) model has been extremely popular in image categorization. The method treats an image as a collection of unordered appearance descriptors extracted from local patches. Then the patches or descriptors are quantized into discrete visual words of a codebook dictionary, and then the image histograms are compared and classified according to the dictionary. The BoF approach discards the spatial order of local descriptors, which severely limits the descriptive power of the image representation. By overcoming this problem, one particular extension of the BoF model, called spatial pyramid matching (SPM) [1], has made a remarkable success on a range of image classification benchmarks and was the major component of the state-of-the-art systems, e.g., [6].

Our method is based on SPM. Similarly like SPM we have used the subdivide and disorder principle. The essence of this principle is to partition the image into smaller blocks and calculate orderless statistics of low level image features. Existing methods differs by the use of features (like pixel value, gradient orientation, and filter bank outputs) and the subdivision method (regular grid, quad trees, and flexible image windows). SPM and as well as our method is independent in choice of features, anyone can plug any other type of features to get a classification result. Authors of [7] offered an early insight into subdivide and principle by suggesting that locally orderless image play an important role in visual perception. While SPM authors did not consider their Gaussian scale space of apertures, we integrated that idea into SPM. Importance of locally orderless statistics is also evident from few recent publications.

To summarize, our method provides a unified framework to combine the gains from subdivide and disorder principle and scale space aperture with a choice of low level features. It will enable to combine the locally orderless statistics results from multiple scales and different fixed hierarchy or rectangular windows to achieve the scene classification task.

2 Related Methods

In this work we combine the power of multiresolution histogram with spatial pyramid matching. So our method consists of two concepts - multiresolution or scale space analysis of image and spatial pyramid matching. In kernel based learning methods like **support vector machine** (SVM), we need to provide a kernel for learning and testing. There are many kernels, which varies in formulation. For example, **histogram intersection kernel** is a kernel matrix which is built by histogram intersection. Essentially it provides a pair wise similarity measure of the training and testing images. A **pyramid match kernel** (PMK) [1] works with an unordered image representation/features. The idea of the method

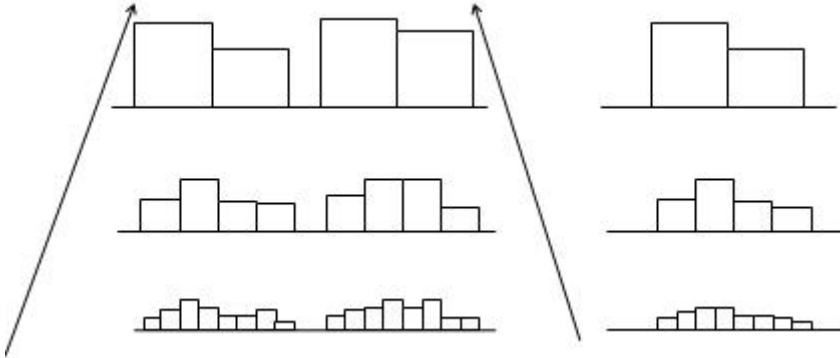


Fig. 1. Schematic illustration of Pyramid match kernel with two levels

is to compute multiresolution histograms and finding the histogram intersection at each resolution. In figure 1, for two different images X and Y , histograms and the corresponding histogram intersections are computed at three resolution levels $(0,1,2)$. The bin size is doubled in successive higher resolutions while the bin numbers are down sampled by 2. After that, all new histogram matching in each resolution is weighted and summed up to form the histogram intersection kernel. It has the limitation of discarding all spatial information. Let us construct a sequence of grids at resolutions $0,1,\dots,L$ such that the grid at level l has 2^l cells along each dimension. Number of matches (I^l) at level l is given by the histogram intersection function. Therefore, the number of new matches found at level l is given by $I^l - I^{l+1}$ for $l = 0,1,\dots,L-1$. The weight associated with level l is set to $\frac{1}{(2^{L-l})}$.

Spatial pyramid matching (SPM) takes a different approach of performing pyramid matching in the two-dimensional image space, and using traditional clustering techniques in feature space. So in SPM the histogram computation is done at a single resolution and in multiple pyramid levels within the same resolution, whereas in PMK it is done in multiresolution. PMK don't employ any feature clustering, directly map features in multiresolution histogram bins. On the other hand, SPM uses feature clustering during histogram computation to find the representative feature sets. In SPM, all feature vectors are first quantized into M discrete types (i.e. the total number of histogram indices is M).

In figure 2 we are showing an example of constructing a three-level spatial pyramid. The image has three types of features, indicated by triangles, circles and stars. At the top row, the image is subdivided at three different levels of resolution. At the bottom row, the number of features that fall in each sub-region is counted. The spatial histograms are weighted according to pyramid match kernel. During kernel computation, each type calculation comprised of two sets of two-dimensional vectors, X_m and Y_m , representing the coordinates of features of type m found in the respective images. The final kernel is then the sum of the separate channel kernels:

$$K^L(X, Y) = \sum_{m=1}^M K^L(X_m, Y_m) \quad (1)$$

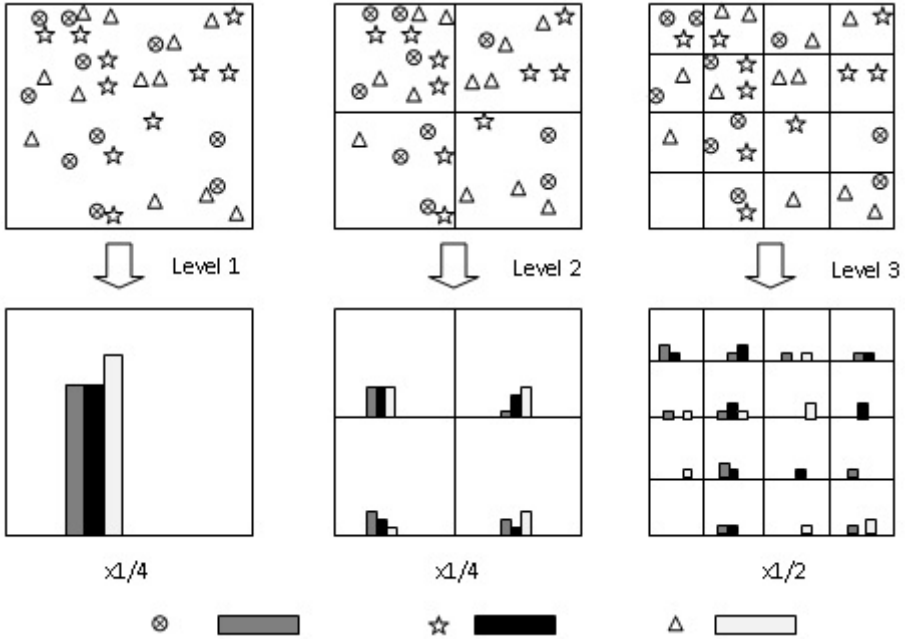


Fig. 2. Three-level spatial pyramid example

This method reduces to a standard bag of features when it is a single level. Considering the fact that pyramid match kernel is simply a weighted sum of histogram intersections, and $c \times \min(a, b) = \min(ca, cb)$ for positive numbers, K^L can be implemented as a single histogram intersection of long vectors formed by concatenating the appropriately weighted histograms of all channels at all resolutions. So essentially we are weighting the histograms before computing the histogram intersection for convenience as the reverse would yield the same result. For L levels and M channels and S scales, the resulting vector has dimensionality:

$$\left(M \sum_{l=1}^L 4^l\right) \times S = M \frac{1}{3} (4^{L+1} - 1) \times S \tag{2}$$

Several experiments reported in results section use the settings of $M = 200$, $L = 3$ and $S = 3$ resulting in (3×17000) -dimensional histogram intersections. However these operations are efficient because the histogram vectors are extremely sparse, the computational complexity of the kernel is linear in the number of features.

One important aspect of the training and test images that we run the experiment only on gray level images; even if color images are available we converted in to gray level images. We decide this from the finding of [9] that removing color information from images doesnt make the scene categorization tasks more attention demanding.

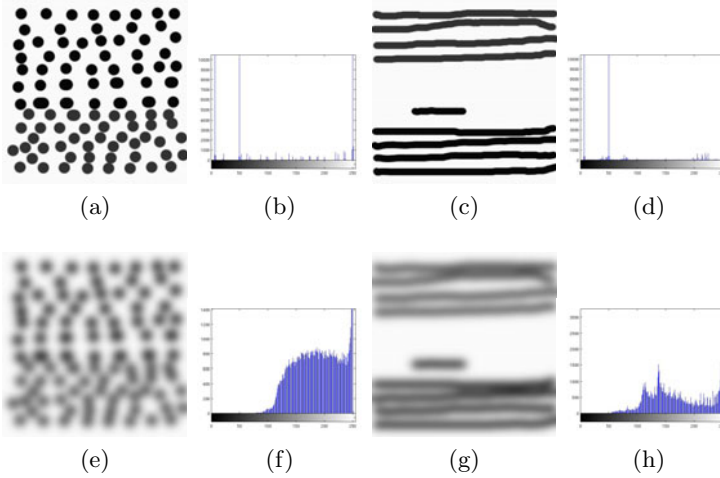


Fig. 3. (a) and (c) are different images with almost similar image histograms (b) and (d). (e) and (g) are corresponding Gaussian blurred images and the previous small difference in histograms is now more prominent in higher scales(f and g).

3 Proposed Method: Multi-scale SPM

SPM uses a mechanism to combine local salient features and their spatial relationship so as to provide a robust feature matching. However, in many cases, different image and block can have similar histograms, this degrade the performance of SPM. This drawback can be overcome by analyzing images in scale space, as confusions in previous case can be clarified at different scales. For example, in figure 3, images (a) and (b) are artificially generated images with almost similar histograms, later they are Gaussian blurred and hence their histograms are also more discriminative than the original histograms. For a given image $f(x,y)$, its linear (Gaussian) scale-space representation is a family of derived signals $L(x,y;t)$ defined by the convolution of $f(x,y)$ with the Gaussian kernel:

$$g_t(x, y) = \frac{1}{2\pi t} e^{-\frac{(x^2+y^2)}{2t}} \quad \text{Such that} \quad L(x, y; t) = (g_t \times f)(x, y) \quad (3)$$

Inspired by scale space theory we want to propose a multi-scale spatial pyramid matching method. Key idea behind our method is the use of scale space to gain more discriminative power in classification. The major steps of our algorithm are (figure 4)

3.1 Feature Generation in Different Scales

First SIFT features are generated from all the images in different scales in a regular grid. Here a dense feature representation is used to avoid the problems superfluous data like clutter, occlusion etc. 128 bit SIFT descriptors are calculated for all images

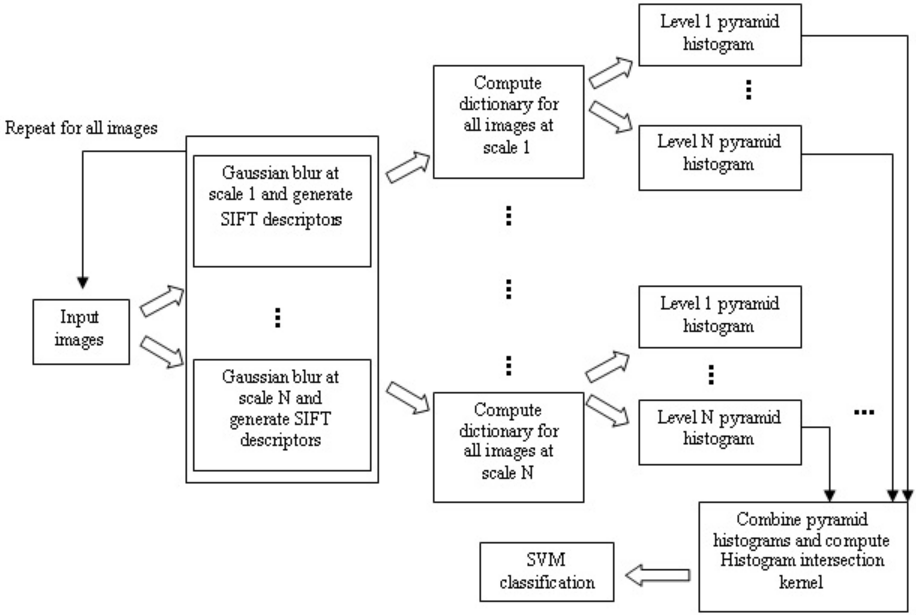


Fig. 4. Block diagram of the proposed method

in all scales in 8*8 regular grid settings and using a 16*16 patch in the grid centers. These features are saved into files for use in later steps.

3.2 Calculate Dictionary

The features are clustered according to the parameter M which is the total number of bins in of the computed histograms. It is often believed that increasing the number of M will increase the classification accuracy. But, in our experiments we are getting comparable accuracy from M=200 setup compared to M=400 and M=600. Again the dictionary is built for all images in all scales. Dictionary is calculated using K-means based clustering using all the extracted SIFT features in a specific scale. In figure 5 (left image), we are showing the corresponding histogram of the values of a 200 sized dictionary. Separate dictionaries are calculated for separate scales. The dictionaries are calculated for using in histogram generation in later stages.

3.3 Compile Pyramid Histogram

For all scales, the image is divided ranging from coarse to finer resolution and compute histogram in each area and assign weight according to PMK. Match in finer resolution will be given more weight than match in coarse resolution. After these steps now we have all the data required to build the pyramid histogram. With the different scale level histograms, we can just concatenate those forming

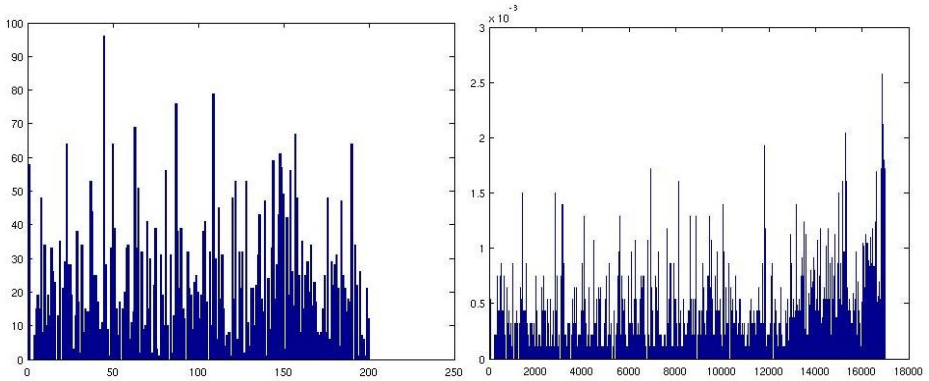


Fig. 5. Histogram plot of the calculated dictionary (left) and combined pyramid histogram plot of all individual histograms in different levels (right)

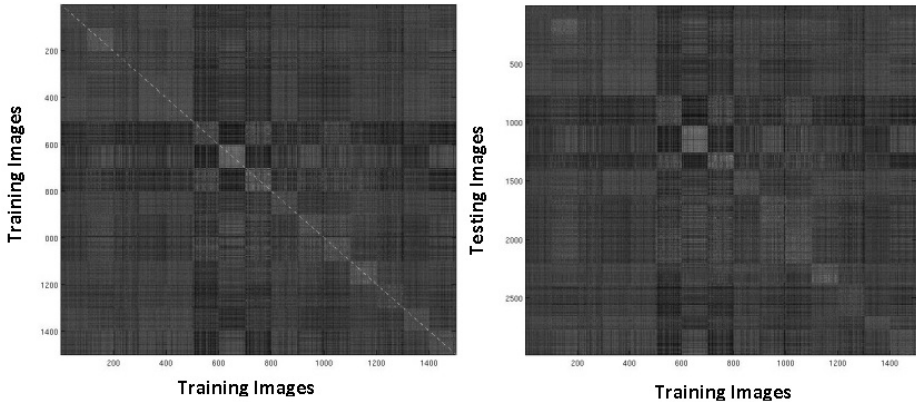


Fig. 6. Histogram intersection kernel as image for Training images (left) and testing images (right)

a long histogram or compute inter-scale intersection/selection before forming the concatenation. We are taking the first approach in our method. Though this will essentially increase the size of the long histogram by the scale factor, but that wouldn't be a problem performance-wise. In this research our focus is on increasing classification accuracy and leveraging performance on the currently available powerful hardware. In figure 5 (right image), one such combined pyramid histogram is shown. According to equation 2, size of the histogram is 34000 for dictionary size 200, 3 pyramid levels and scale level 1.

3.4 Kernel Computation and SVM Classification

For SVM, we just need to build the histogram intersection kernel from the compiled pyramid histograms. As we explained before, for the histogram intersection kernel computation we just need to find the intersections of the long histogram

Table 1. Statistical information of the image datasets used

Dataset	No. of categories	Total No. of images	Avg. image size	Max. no. of train/test images used
Scene category	15	4485	300*250	100/rest
Caltech-101	102	9144	300*200	30/300
Caltech-256	257	30607	351*300	60/300

concatenation formed in the previous step. For training kernel intersection is computed between the same concatenated histograms and for training kernel it is between training histogram and testing histogram. A grey scale image map of the testing and training kernel is shown in figure 6. For training kernel, a white line is visible along the diagonal, as there will be a perfect match for corresponding training pairs. In testing kernel the matches are scattered as training and testing sets are different. For SVM, we are using a modified version of libSVM library [10] which implements the one vs. all classification. scales and different fixed hierarchy or rectangular windows to achieve the scene classification task.

4 Experimental Results

4.1 Test Dataset

We tested our method on scene category dataset [1], Caltech-101 [11] and Caltech-256 [12]. A brief statistical comparison of these three datasets is given in table 1.

4.2 Performance Metric

Two separate performance metric is used to measure the results combined accuracy and average of per class accuracy. **Per class accuracy** (P) is defined as the ratio of correctly classified images in a class with respect total number of images in that particular class. If total number of image categories is N, then combined accuracy and average of per class accuracy is defined as:

$$\text{Average of per class accuracy} = \frac{\sum_{i=1}^N P_i}{N} \quad (4)$$

$$\text{Combined accuracy} = \frac{\text{Total number of correctly classified images} \times 100}{\text{Total number of images in the dataset}} \quad (5)$$

Table 2 is the extensive experiment done with codebook size, pyramid level, scale level. Results are first grouped by codebook size and pyramid levels. The notable thing here is that, scale level greater than one always produce better results than single level. Using the combined accuracy metric, we get our best result from codebook size 400, pyramid level 3 and scale level 2. Scale level 1 is basically the original SPM. So for scale level 1, we use the results from [1]. But

Table 2. Accuracy results on different combination of parameters. Bold font means its the best for a certain codebook size and pyramid level.

Codebook Size	Pyramid level	Scale level	Combined accuracy (%)	Avg. of per class accuracy (%)
200	3	1	81.47 ± 0.59	81.11 ± 0.68
200	3	2	83.69 ± 0.50	83.31 ± 0.59
200	3	3	83.45 ± 0.57	83.21 ± 0.61
200	2	1	79.88 ± 0.52	81.1 ± 0.30
200	2	2	82.69 ± 0.67	82.25 ± 0.52
200	2	3	82.78 ± 0.70	82.21 ± 0.75
400	3	1	81.95 ± 0.57	81.1 ± 0.60
400	3	2	83.78 ± 0.64	83.48 ± 0.58
400	3	3	83.71 ± 0.54	83.29 ± 0.70
400	2	1	80.28 ± 0.53	81.4 ± 0.50
400	2	2	83.22 ± 0.44	82.75 ± 0.40
400	2	3	83.10 ± 0.63	82.67 ± 0.78

Table 3. Our result compared to the original SPM for codebook size = 400, pyramid level = 3 and scale level = 2

	SPM [1]	Proposed method
Average of per class accuracy(%)	81.1 ± 0.60	83.48 ± 0.58
Combined accuracy(%)	81.95 ± 0.57	83.78 ± 0.64

Table 4. Caltech-101 result for codebook size = 400, pyramid level = 3 and scale level = 3

	SPM [1]	Proposed method
Average of per class accuracy(%)	64.6 ± 0.7	67.36 ± 0.17
Combined accuracy(%)	70.59 ± 0.16	76.65 ± 0.46

Table 5. Caltech-256 result for codebook size = 400, pyramid level = 3 and scale level = 3

	SPM [12]	Proposed method
Average of per class accuracy(%)	32.62 ± 0.41	37.54 ± 0.31
Combined accuracy(%)	34.98 ± 0.60	40.19 ± 0.12

as the authors of [1] didnt report the result of combined accuracy, we calculated it using our own implementation of SPM. All results are obtained using a 2*64 bit Quad core processor with 48 GB of RAM. All experiments are run for ten times with randomly selected training and testing images. The average of all the

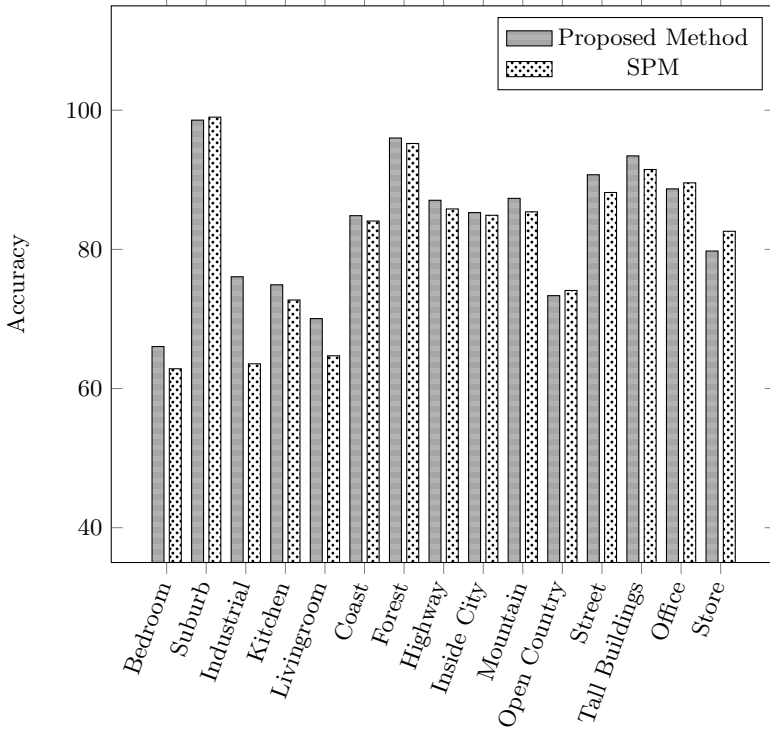


Fig. 7. Per class accuracy for the result (average of per class accuracy) reported in Table 2

runs and standard deviation is reported here. Table 3 summarizes our best result compared to the original SPM. In figure 7, we showed the per class accuracy for the best result reported in Table 4. Our method outperforms SPM in eleven categories and provides comparable performance in the four categories. We tested whether the difference between two methods reported in table 2 is statistically significant by the Matlab function `ttest`. In this case, `ttest` result indicated that the improvement obtained by the proposed method is indeed statistically significant. The results on Caltech-101 and Caltech-256 are presented in table 4, 5 and it is in line with the results obtained from scene category dataset. On both of these databases, according to overall average accuracy metric, proposed method is better than SPM by around 3% margin and using the average of per class accuracy metric, the margin is around 6%.

5 Conclusion and Future Scope

This paper presents an improvement to the spatial pyramid matching scheme. We provided a simple, intuitive and effective way to improve the SPM method. To the best of our knowledge, this has not been done by previous researchers.

The proposed extension is quite general and not limited to any specific feature descriptors or classifiers and can be used as a surrogate module or new baseline for SPM in image categorization systems.

The weight mechanism of the spatial pyramid matching (SPM) method is not sophisticated enough. It defines uniform and better weight level to the finer resolution blocks and punishes the coarse resolution blocks by assigning less weight. As a basic method this is okay, but consider a finer resolution block containing only background or clutter, then assigning it more weight is only misleading calculation. So in the future, there is room for redesigning this weight mechanism to only assigning more weight to the corresponding blocks irrespective of scale or spatial resolution.

References

1. Lazebnik, S., Schmid, C., Ponce, J.: Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 2169–2178 (2006)
2. Oliva, A., Torralba, A.: Modeling the shape of the scene: a holistic representation of the spatial envelope. *International Journal of Computer Vision* 42(3), 145–175 (2001)
3. Ogel, J., Schwaninger, A., Wallraven, C., Bühlhoff, H.H.: Categorization of Natural Scenes: Local versus Global Information and the Role of Color. *ACM Transactions on Applied Perception* 4(3) (2007)
4. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* 60(3), 91–110 (2004)
5. Witkin, A.P.: Scale-space filtering. In: Proceedings of 8th International Joint Conference on Artificial Intelligence, pp. 1019–1022 (1983)
6. Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge. In: VOC 2009 (2009), <http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html>
7. Koenderink, J., Doorn, A.V.: The structure of locally orderless images. *International Journal of Computer Vision* 31(199), 159–168
8. Grauman, K., Darrell, T.: The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features. In: Proceedings of the IEEE International Conference on Computer Vision, ICCV (2005)
9. Fei-fei, L., Perona, P.: A Bayesian hierarchical model for learning natural scene categories. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2005)
10. Chang C., Lin C.: LIBSVM: a library for support vector machines (2001), <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
11. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In: Proceedings of IEEE Workshop on Generative-Model Based Vision, CVPR (2004)
12. Griffin, G., Holub, A., Perona, P.: Caltech-256 Object Category Dataset. Caltech Technical Report. Technical Report, Caltech (2007)

Dense Multi-frame Optic Flow for Non-rigid Objects Using Subspace Constraints

Ravi Garg¹, Luis Pizarro², Daniel Rueckert², and Lourdes Agapito¹

¹ Queen Mary University of London, Mile End Road, London E1 4NS, UK

² Imperial College London, South Kensington Campus, London SW7 2AZ, UK

Abstract. In this paper we describe a variational approach to computing dense optic flow in the case of non-rigid motion. We optimise a global energy to compute the optic flow between each image in a sequence and a reference frame simultaneously. Our approach is based on subspace constraints which allow to express the optic flow at each pixel in a compact way as a linear combination of a 2D motion basis that can be pre-estimated from a set of reliable 2D tracks. We reformulate the multi-frame optic flow problem as the estimation of the coefficients that multiplied with the known basis will give the displacement vectors for each pixel. We adopt a variational framework in which we optimise a non-linearised global brightness constancy to cope with large displacements and impose homogeneous regularization on the multi-frame motion basis coefficients. Our approach has two strengths. First, the dramatic reduction in the number of variables to be computed (typically one order of magnitude) which has obvious computational advantages and second, the ability to deal with large displacements due to strong deformations. We conduct experiments on various sequences of non-rigid objects which show that our approach provides results comparable to state of the art variational multi-frame optic flow methods.

1 Introduction

Dense registration of deforming surfaces from a monocular image sequence continues to be one of the unsolved fundamental problems in computer vision, despite the attention it has received from the community for decades. Its applications are numerous from video augmentation to non-rigid structure from motion or medical imaging.

For instance, non-rigid structure from motion [1,2,3] and scene-flow techniques for deformable surfaces [4] rely on the efficient estimation of image correspondences. However, most state of the art algorithms only use sparse features obtained with local feature matching techniques such as Lucas and Kanade’s popular tracker [5]. While this algorithm provides good matches in areas with rich texture, it fails to provide reliable solutions in texture-less areas with vanishing gradients due to the well known aperture problem. Therefore, most current non-rigid structure from motion algorithms are limited to sparse 3D reconstructions.

Irani’s was the first work to exploit rank constraints in the case of rigid objects to obtain optic flow in areas with one dimensional or no texture [6]. She proved

that the optic flow vectors lie on a lower dimensional subspace and therefore the flow at each point can be expressed as a linear combination of a low-rank motion basis. This constraint was then used for estimating dense correspondences, by requiring that all corresponding points across all video frames reside in the appropriate low-dimensional linear subspace. However, although this algorithm indeed allows to solve the aperture problem, it performs poorly in areas of uniform intensity. Moreover, this approach cannot cope with large displacements since they rely on the linearised brightness constancy assumption which assumes small displacements. Besides, since they do not impose smoothness constraints, the resulting optic flow is not regular.

The rank constraint was later extended to the non-rigid case by Torresani *et al.* [7] and Brand [8]. These methods also minimise the linearised brightness constancy and therefore they suffer when there are image displacements larger than a few pixels or local appearance changes due to large deformations. Besides, although in theory these non-rigid approaches are dense, in practice they have only been used to extend the tracking to features which display the aperture problem (such as edges or degenerate features) instead of computing optic flow values for every pixel in the image.

In contrast, variational methods allow to formulate the optic flow problem in its continuous form. Pioneered by Horn and Schunck [9], the optic flow problem is formulated as the optimization of an energy functional with a regulariser that allows to fill in textural information into non-textured regions from their neighbourhoods, making dense flow field estimation possible. Moreover, recent developments in variational optical flow [10,11] have proposed numerical strategies to solve the *non-linearised* brightness constancy constraint in order to cope with large displacements.

Although geometric constraints have been incorporated before in the computation of optic flow, this has been in the case of rigid scenes. For instance, fundamental matrix priors have been proposed within variational approaches [12,13,14] to improve the accuracy of optic flow. To the best of our knowledge this is the first approach to apply subspace constraints to the case of non-rigid motion in a variational framework.

In this paper we propose to marry the ideas of using subspace constraints to constrain the optic flow and solving the *non-linearised* brightness constancy constraint within a variational approach. This allows us on the one hand to reduce the dimensionality of the multi-frame optic flow problem drastically and on the other to be able to cope with large displacements while imposing smoothness on the optic flow taking advantage of the variational formulation. We focus on the more challenging problem of non-rigid motion and adopt a multi-frame optical flow approach by optimising a global energy.

2 Low-Rank Non-rigid Subspace Constraints

Our approach is based on the assumption that the motion of a non-rigid object can be represented as a linear combination of K basis shapes which encode the mean shape and its main modes of deformation. This low-rank constraint, first proposed

by Bregler *et al.* [15], has allowed the simultaneous estimation of 3D non-rigid shape and motion. More importantly for the scope of this paper, it also induces a subspace rank constraint on the 2D motion of image points. In the next two sections we describe the linear basis shape model that underpins our formulation and the linear subspace constraint satisfied by the multi-frame correspondences.

2.1 Linear Basis Shape Model

The 3D reconstruction of a rigid body from a monocular sequence under the affine projection assumption was pioneered by Tomasi and Kanade in [16] using a factorization approach. This assumption was then relaxed to extend structure from motion algorithms to the case of deformable objects. Most non-rigid structure from motion algorithms are based on the low rank basis shape model defined by Bregler *et al.* [15] in which the deformable 3D shape is represented as a linear combination of a set of basis shapes with time varying coefficients.

In the case of deformable objects the observed 3D points change as a function of time. In this paper we use the low-rank shape model defined by Bregler *et al.* [15] in which the P observed 3D points deform as a linear combination of a fixed set of K rigid shape bases according to time varying coefficients. In this way, $\mathbf{S}_f = \sum_{k=1}^K l_{fk} \mathbf{B}_k$ where the matrix $\mathbf{S}_f = [\mathbf{S}_{f1}, \dots, \mathbf{S}_{fP}]$ is the 3D shape of the object at frame f , the $3 \times P$ matrices \mathbf{B}_k are the shape bases and l_{fk} are the coefficient weights. If we assume an orthographic projection model, the coordinates (x_{fj}, y_{fj}) of the 2D image points j observed at frame f are then related to the coordinates of the 3D points according to the following equation:

$$\hat{\mathbf{W}}_f = \begin{bmatrix} x_{f1} & x_{f2} & \dots & x_{fP} \\ y_{f1} & y_{f2} & \dots & y_{fP} \end{bmatrix} = \mathbf{R}_f \left(\sum_{k=1}^K l_{fk} \mathbf{B}_k \right) + \mathbf{T}_f \tag{1}$$

where \mathbf{R}_f is a 2×3 truncated rotation matrix and the $2 \times p$ matrix \mathbf{T}_f aligns the image coordinates to the image centroid. When the image coordinates are registered to the centroid of the object and we consider all the frames in the sequence, we may write the measurement matrix as:

$$\hat{\mathbf{W}}_f = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1P} \\ y_{11} & y_{12} & \dots & y_{1P} \\ \vdots & \vdots & \vdots & \vdots \\ x_{F1} & x_{F2} & \dots & x_{FP} \\ y_{F1} & y_{F2} & \dots & y_{FP} \end{bmatrix}_{2F \times P} = \begin{bmatrix} l_{11} \mathbf{R}_1 & \dots & l_{1K} \mathbf{R}_1 \\ \vdots & \ddots & \vdots \\ l_{F1} \mathbf{R}_F & \dots & l_{FK} \mathbf{R}_F \end{bmatrix} \begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_K \end{bmatrix} = \mathbf{M}_{2F \times 3K} \mathbf{B}_{3K \times P} \tag{2}$$

Since \mathbf{M} is a $2F \times 3K$ matrix and \mathbf{B} is a $3K \times P$ matrix, in the case of deformable structure the rank of $\hat{\mathbf{W}}_f$ is constrained to be at most $3K$. This rank constraint forms the basis of the factorization method for the estimation of 3D deformable structure and motion [15]. Interestingly, the motion and shape matrices can exchange their roles as basis and coefficients and we can either interpret the 2D tracks as the projection of a linear combination of 3D basis shapes (\mathbf{B}_k) or as the

linear combination of a 2D motion basis encoded in matrix \mathbf{M} . This concept of 2D trajectory basis was introduced by Torresani *et al.* [17] as an extension to the non-rigid case of the subspace constraints proposed Irani [6].

2.2 2D Low-Rank Trajectory Basis

Let us denote the 2D motion of point j with respect to its position in the reference image $\mathbf{x}_{0j} = (x_{0j}, y_{0j})$ by the vector

$$\mathbf{w}_j = [u_{1j} \ u_{2j} \ \dots \ u_{Fj} | v_{1j} \ v_{2j} \ \dots \ v_{Fj}]^T \tag{3}$$

where $u_{ij} = x_{ij} - x_{0j}$ and $v_{ij} = y_{ij} - y_{0j}$. We now consider P pixels or image features observed in the image and stack their corresponding multi-frame 2D motion vectors horizontally to form a $2F \times P$ measurement matrix \mathbf{W}

$$\mathbf{W} = [\mathbf{w}_1 \ \mathbf{w}_2 \ \dots \ \mathbf{w}_P] \tag{4}$$

It is easy to observe that \mathbf{W} is also rank constrained, therefore the multi-frame motion vector \mathbf{w}_j of any point j can be expressed as linear combination of R basis motion vectors \mathbf{Q}_r

$$\mathbf{w}_j = \sum_{r=1}^R L_j^r \mathbf{Q}_r \tag{5}$$

where $\mathbf{Q}_r = [Q_{1r}^u \ Q_{2r}^u \ \dots \ Q_{Fr}^u | Q_{1r}^v \ Q_{2r}^v \ \dots \ Q_{Fr}^v]^T$ and L_j^r with $r = 1 \dots R$ are the coefficients that multiply basis vectors Q_r to obtain \mathbf{w}_j . We may now rewrite this equation for every point in matrix form as

$$\mathbf{W}_{2F \times P} = \underbrace{\begin{bmatrix} Q_{11}^u & \dots & Q_{1r}^u & \dots & Q_{1R}^u \\ \vdots & & \vdots & & \vdots \\ Q_{F1}^u & \dots & Q_{Fr}^u & \dots & Q_{FR}^u \\ \hline Q_{11}^v & \dots & Q_{1r}^v & \dots & Q_{1R}^v \\ \vdots & & \vdots & & \vdots \\ Q_{F1}^v & \dots & Q_{Fr}^v & \dots & Q_{FR}^v \end{bmatrix}}_{\mathbf{Q}_{2F \times R}} \underbrace{\begin{bmatrix} L_1^1 & \dots & L_j^1 & \dots & L_P^1 \\ L_1^2 & \dots & L_j^2 & \dots & L_P^2 \\ \vdots & & \vdots & & \vdots \\ L_1^R & \dots & L_j^R & \dots & L_P^R \end{bmatrix}}_{\mathbf{L}_{R \times P}} = \begin{bmatrix} \mathbf{Q}^u \\ \mathbf{Q}^v \end{bmatrix} \mathbf{L} \tag{6}$$

It is easy to see that the motion basis matrix \mathbf{Q} is independent of the number of points. Therefore it is possible to pre-compute \mathbf{Q} from a small subset of “reliable” point tracks by truncating the *SVD* decomposition of the corresponding measurement matrix \mathbf{W}_{rel} to rank R .

$$svd(\mathbf{W}_{rel}) = \underbrace{\mathbf{U}}_{\mathbf{Q}} \underbrace{\mathbf{\Lambda V}^T}_{\mathbf{L}_{rel}} \tag{7}$$

The “reliable” tracks are those where the texture of the image is strong in both spatial directions which can be selected using Shi *et al.* [18]. Now the motion of any other point, or set of points, in the image can be encoded in terms of the known basis \mathbf{Q} and only the new coefficients \mathbf{L} need to be computed. This will form the basis of our re - parametrisation of the multi-frame optic flow problem.

In the next section we will introduce our variational approach to solve the multi-frame dense optic flow problem for non-rigid motion using rank constraints.

3 Dense Optic Flow with Subspace Constraints

As we mentioned above, current non-rigid structure from motion approaches are limited to sparse 3D reconstructions as they rely on sparse feature tracking techniques such as the method of Lucas and Kanade [5]. This local method copes with the well known *aperture problem* in areas rich of texture, but fails to estimate motion in (flat) regions with vanishing image gradients. This limitation was overcome by the pioneered work of Horn and Schunck [9] who formulated a global energy functional with a regulariser that allows to fill in textural information into flat regions from their neighbourhoods, making dense flow field estimation possible. More advanced methods for dense motion recovery have been proposed in the literature. Amongst them, the *combined local and global* (CLG) approach of Bruhn *et al.* [19] casts the methods of Lucas-Kanade and Horn-Schunck into a unifying variational formulation; and the *large displacement optical flow* (LDOF) method of Brox and Malik [20] integrates rich feature descriptors into a variational optic flow approach.

In this section we aim at combining dense motion estimation with the subspace rank constraints described in the previous section following variational principles. As this is the first attempt of that kind -to the best of our knowledge-, we embed the rank constraints in the original approach of Horn and Schunck as a proof of concept that subspace constraints can be used to determine multi frame optic flow. Let $I_0, I_f : \Omega \in \mathbb{R}^2 \rightarrow \mathbb{R}$ be the reference and the target images to be registered (or matched). To compute the optic flow $(u(\mathbf{x}), v(\mathbf{x}))^\top$ for all $\mathbf{x} := (x, y)^\top \in \Omega$, the Horn-Schunck [9] method minimises an energy functional of the form $E = E_{data} + \alpha E_{reg}$:

$$E(u, v) = \int_{\Omega} \left((I_f(x + u, y + v) - I_0(x, y))^2 + \alpha(|\nabla u|^2 + |\nabla v|^2) \right) d\mathbf{x}, \quad (8)$$

where E_{data} and E_{reg} penalise deviations from the model assumptions and $\alpha > 0$ acts as a regularisation parameter. The assumption in E_{data} is that the grey value of a “moving” pixel remains constant in both images, while E_{reg} assumes that the optic flow varies smoothly in space. It is important mentioning that since we are dealing with a sequence, we adopt a *non-linearised* data constraint in order to cope with large displacements following [10, 11]. This choice has the disadvantage that the energy functional may be non-convex, and hence with multiple local minima. We discuss how to alleviate this problem later in this section.

3.1 Horn-Schunck Approach with Subspace Constraints

We now extend the variational model (8) for tracking a non-rigid object along an image sequence I_1, \dots, I_F using subspace constraints. Following the derivations from Section 2.2, we can express the optic flow between a reference image I_0 and a target image I_f as a linear combination of 2D motion basis

$$u_f(\mathbf{x}) = \mathbf{Q}_f^u \mathbf{L}(\mathbf{x}), \quad (9)$$

$$v_f(\mathbf{x}) = \mathbf{Q}_f^v \mathbf{L}(\mathbf{x}), \quad (10)$$

which holds for all frames $f = 1, \dots, F$. The $(1 \times R)$ -vectors \mathbf{Q}_f^u and \mathbf{Q}_f^v correspond to the f -th row of the motion matrix basis \mathbf{Q}^u and \mathbf{Q}^v respectively. Assuming that the motion basis can be pre-estimated using a set of reliable 2D tracks, we reformulate the optic flow computation as the estimation of the $(R \times 1)$ -vector of flow coefficients $\mathbf{L}(\mathbf{x})$, for all $\mathbf{x} \in \Omega$. The advantage of this parameterisation is that the functions $\mathbf{L}(\mathbf{x})$ are shared by both components of the flow along the whole image sequence, which drastically reduces the number of unknowns. Therefore, using the above parameterisation we proposed the following extension of the Horn-Schunck approach for the dense multi-frame estimation of the functions \mathbf{L} :

$$E(\mathbf{L}) = \int_{\Omega} \sum_{f=1}^F (I_f(x + \mathbf{Q}_f^u \mathbf{L}(\mathbf{x}), y + \mathbf{Q}_f^v \mathbf{L}(\mathbf{x})) - I_0(x, y))^2 dx + \alpha \int_{\Omega} \sum_{f=1}^F (|\nabla(\mathbf{Q}_f^u \mathbf{L}(\mathbf{x}))|^2 + |\nabla(\mathbf{Q}_f^v \mathbf{L}(\mathbf{x}))|^2) dx. \tag{11}$$

By noticing that the motion basis matrix \mathbf{Q} does not vary spatially (cf. (6)) and that its column-vectors are orthonormal (cf. (7)), the smoothness constraint in (11) can be simplified to $\sum_{r,f} (Q_{f,r}^u{}^2 + Q_{f,r}^v{}^2) |\nabla L^r(\mathbf{x})|^2 = \sum_r |\nabla L^r(\mathbf{x})|^2$, where $L^r(\mathbf{x})$ is the r -th element of $\mathbf{L}(\mathbf{x})$. This means that we are actually imposing homogeneous regularisation on the multi-frame motion basis coefficients. With that simplification, the proposed energy functional reads

$$E(\mathbf{L}) = \int_{\Omega} \left(\sum_{f=1}^F (I_f(x + \mathbf{Q}_f^u \mathbf{L}(\mathbf{x}), y + \mathbf{Q}_f^v \mathbf{L}(\mathbf{x})) - I_0(x, y))^2 + \alpha \sum_{r=1}^R |\nabla L^r(\mathbf{x})|^2 \right) dx. \tag{12}$$

Before discussing the minimisation strategy for this energy, it is important to state that once the functions \mathbf{L} have been estimated we can densely compute the optic flow between all target images I_1, \dots, I_F and the reference image I_0 via the equations (9)-(10). Thanks to the non-linearised grey value constancy assumption we can cope with large displacements, which we will demonstrate in the experimental section. This is one of the characteristics that distinguishes our approach from other methods such as [6,7] that assume small deformations as they rely on linearised data constraints. Moreover, by exploiting a global variational formulation we can truly compute dense and smooth flow fields without worrying about the aperture problem in areas of partial (one-dimensional) or null textural information.

One could argue that in (12) a pre-computed motion basis needs to be available before estimating \mathbf{L} . This is though a common and very useful practice in numerous applications where the tracked objects undergo particular motions that could be represented by suitable bases. Such bases are frequently computed from sparse data points, so the need for estimating the motion coefficients and then the optic flow densely is compelling. Our approach is going in that direction, and in the future we expect to jointly estimate the motion basis and the motion coefficients

for specific applications. In terms of dimensionality reduction, the number of unknowns in the functional (12) is $R \times P$, where R is the number of 2D motion bases and P the number of pixels, compared to the $2 \times F \times P$ unknowns in a multi-frame optical flow estimation. This way we reduce the number of variables by a factor of $2F/R$ with $F \gg R$. In a typical experiment we would consider sequences with around $F = 50$ frames and use an average of $R = 5$ basis.

3.2 Minimisation

The proposed energy functional $E(L)$ from (12) can be minimised by solving the Euler-Lagrange equations (21) for all $r = 1, \dots, R$

$$0 = \sum_{f=1}^F \left(I_{fz} (Q_{f,r}^u I_{fx} + Q_{f,r}^v I_{fy}) - \alpha \Delta(L^r) \right), \quad (13)$$

with reflecting boundary conditions, where $(I_{fx}, I_{fy})^\top := \nabla I_f$, $I_{fz} := I_f - I_0$ and $\Delta := \partial_{xx} + \partial_{yy}$ is the Laplacian operator.

As we mentioned above the energy (12) may be non convex due to the non-linearised data constraint. For the same reason, we obtain a highly non linear system of equations (13). Following (11) we solve the system of equations by the following multiresolution strategy with warping to avoid local minima and handle large displacements, discretising semi-implicitly the data term and fully implicit the regularisation term

$$0 = \sum_{f=1}^F \left(I_{fz}^{k+1} (Q_{f,r}^u I_{fx}^k + Q_{f,r}^v I_{fy}^k) - \alpha \Delta(L^{r^{k+1}}) \right), \quad (14)$$

with k the warping index for a pyramid level k iteration. We remove the nonlinearities cause by I_{fz} via the Taylor expansion

$$I_{fz}^{k+1} = I_{fz}^k + \sum_{\eta=1}^R \{ I_{fx}^k Q_{f,\eta}^u + I_{fy}^k Q_{f,\eta}^v \} dL^{\eta^k}, \quad (15)$$

where $dL = (dL^1, \dots, dL^R)^\top$. We discretise the partial derivatives with standard finite differences and solve the resulting linear system of equations with the SOR solver (22).

4 Experimental Results

We have evaluated our approach on three different video sequences which display different types of deformations. In all three sequences we used the technique of Lucas and Kanade (5) to track a set of highly textured reliable points. The motion basis was in each case estimated by computing the singular value decomposition of the measurement matrix containing the reliable tracks and truncating to the chosen rank. Since these R basis tracks must encode the motion of any other pixel

in the image, we are implicitly assuming that the reliable tracks cover the object and they represent a good sample of the deformations present in the sequence. Despite having significant texture, our sequences are challenging mainly because of the presence of complex deformations and large displacements.

It is important to stress that we were not able to test our algorithm on the Middlebury database simply because there were no deformable motion sequences. Therefore we have not been able to test our algorithm on ground truth data. However, we provide various qualitative tests and a comparison with a state of the art variational method for large displacements [20].

Paper sequence: The first sequence shows a sheet of paper being bent backwards and has been used in non-rigid structure from motion methods [23] for 3D reconstruction. We used a 40 frame long subsequence and tracked corner features to estimate the basis. In this sequence we chose the rank to be 2 and considered only the first 2 basis tracks. Since the sequence had 40 frames, the number of variables to be estimated per pixel in the variational framework was reduced from 80 to just 2.

We show results in Figure(I). The reference frame and two other frames of the sequence are shown on the top row. These show the extent of the deformation and the large displacements (a maximum of 58 pixels). The left column shows the reference image on top and the results of reverse warping ($W^{-1}(I_f)$) the other two frames to the reference frame. These results show that the algorithm can cope with the challenging displacements present in this sequence. The two rightmost plots in the middle row of Figure(I) show Middlebury colour coded optic flow results for every pixel. The colour indicates direction and the intensity indicates magnitude of the flow. These results prove the smoothness of our results. Finally, we have sampled the dense optic flow values and show the arrow values on those sampled locations. These plots also confirm the smoothness and the accuracy of the resulting optical flow.

Face Paint sequence: We use a face paint sequence provided by an artist¹ which has strong and fast deformations. Besides there is significant appearance change in most of the local features, further challenging our system. This is obvious by looking at the top row of Figure(2) which shows three images of the sequence (including the first/reference and last). In this case we chose a 40 frame long subsequence and needed as few as 4 basis tracks to encode the non-rigid motion. Figure(2) shows the results. Despite being a challenging sequence with self-occlusions (for instance in the eye area), large displacements of up to 27 pixels and large appearance changes, the reverse warped images appear to be accurate and the colour coded optic flow results show smoothness in the flow.

T-shirt sequence: This particular sequence has no motion blur, self or external occlusions but still has highly non-rigid and large deformations. We only found 90 reliable features with Lucas and Kanade's tracking algorithm on the 60 frames of the sequence where a T-shirt is coming back to its rest state from a deformed one. The maximum displacement in this sequence was 50 pixels. In this case, we

¹ This sequence is courtesy of James Kuhn.

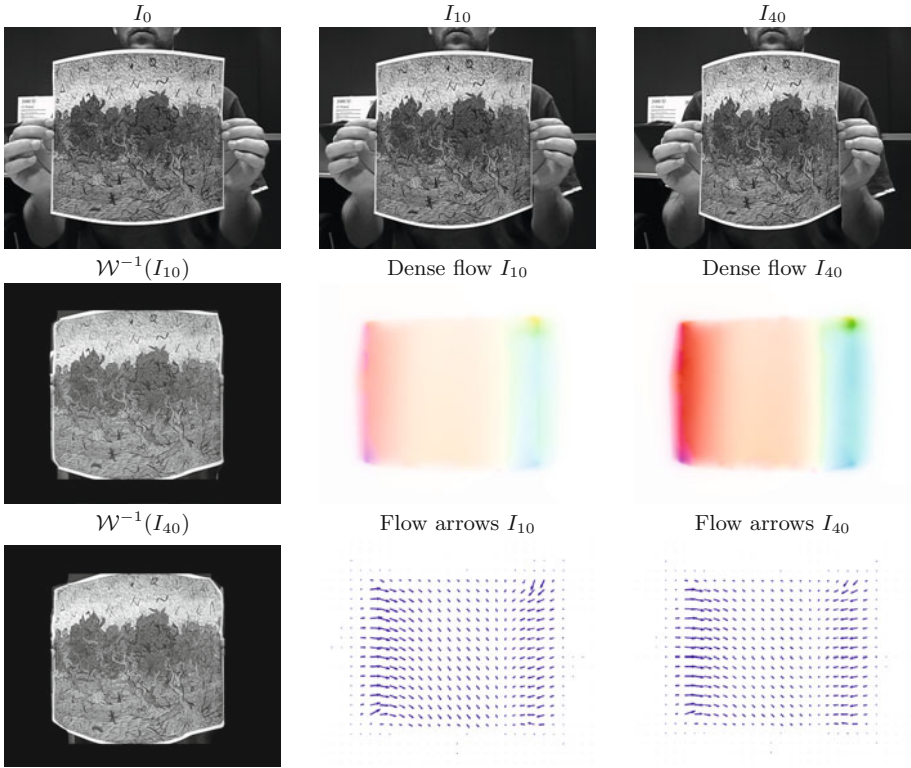


Fig. 1. Results for the *Paper sequence*. Top row shows the reference image and two more images in the sequence (rightmost is the last frame). The left-most column shows the reference image on top and the result of reverse warping images I_{10} and I_{40} to the reference frame. Colour coded images represent the dense optical flow between each image and the reference frame. Arrows represent sampled flow vectors between each image and the reference frame.

only needed 3 motion basis to run our variational multi-frame optic flow. Our results are shown in Figure(3). The bottom left image shows the result of reverse warping the last image of the sequence back to the first/reference frame. There are some obvious errors in the corner of bottom corner of the t-shirt. However, this was expected since there is no texture there and therefore no features were found in that area to encode in the basis. The rest of the results show smooth flow with good accuracy, except in the corners where no features were tracked. The results follow the same layout as Figure(1).

4.1 Comparison with State of Art Variational Optic Flow Method

Comparing our approach with other multi-frame optic flow algorithms is not straightforward. Our problem definition is different from other approaches since we compute the optic flow between a reference frame and every other frame in the sequence with subpixel accuracy. Note that the flow between the first

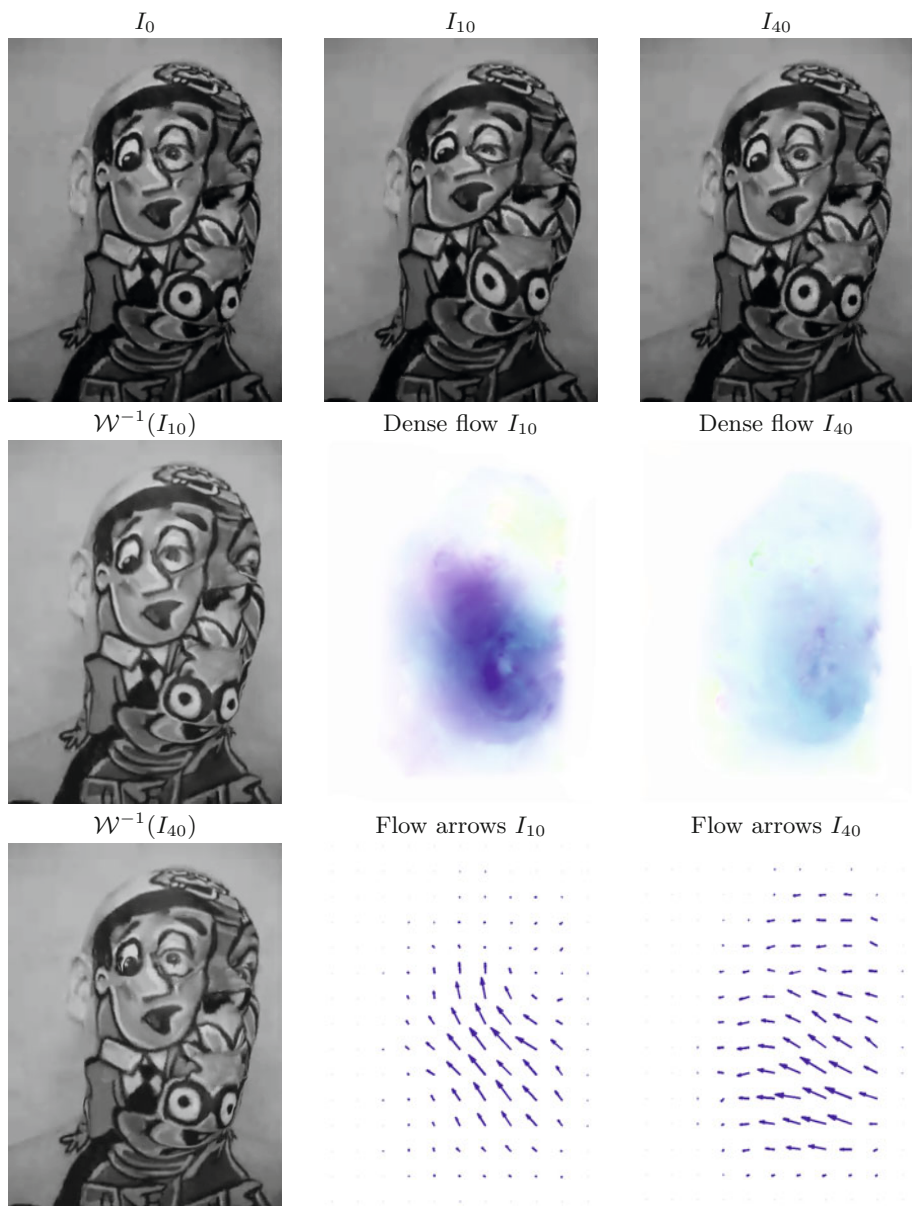


Fig. 2. Results for the frame 10 and frame 40 of *Face-paint* sequence with same layout as figure 1

and last image will be very large. Other approaches tend to compute frame-to-frame flow with subpixel accuracy but in their case it is not possible to register the last frame to the reference without the use of interpolation. Therefore it is difficult to carry out a fair comparison. It is also worth mentioning that, unlike

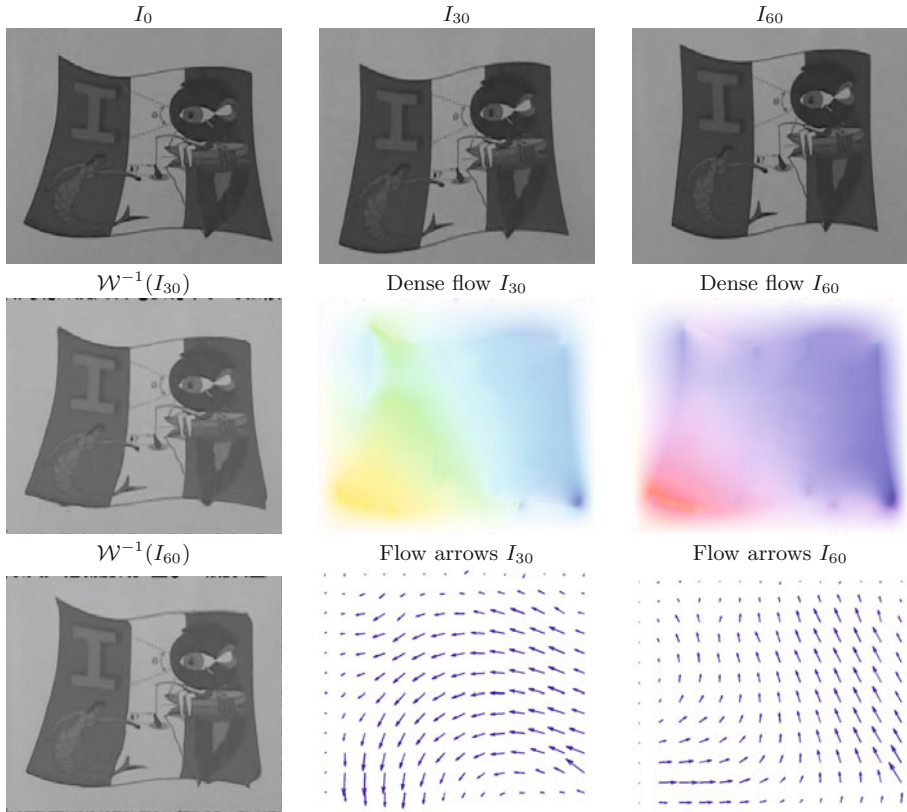


Fig. 3. Results for the frame 10 and frame 40 of *T-shirt sequence* with same layout as figure 1

other authors [24], despite using image sequences in our experiments we do not assume temporal smoothness in optic flow. We only enforce spatial smoothness of the flow, which leads to smoothing of the motion basis coefficients.

We have chosen to compare the performance of our algorithm with Brox and Malik’s large displacement optical flow (LDOF) [20] which integrates rich feature descriptors into a variational optic flow approach to compute dense flow. This approach can be considered the state of the art in the case of very large displacements, since it outperforms previous methods. Although both the data term and the regulariser are more advanced than the ones we have used in our variational formulation we thought it fair to compare our approach with the best performing method for large displacements, particularly since it integrates the use of features.

We compute the flow between the reference frame and frame 10 of the face sequence using LDOF [20] and compare the results with our multi-frame approach. Figure (4) shows the detailed comparison. Note that we only used 60 Lucas-Kanade features in this case. The left-most images show the target image

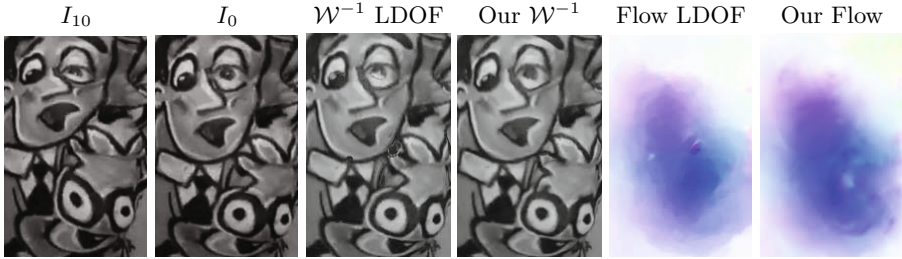


Fig. 4. Left-most two: Target and reference images. Middle two: results of reverse warping target image into reference frame with LDOF (left) and our approach (right). Right two: colour coded optical flow.

I_{10} and the reference frame I_0 . The two middle images show the results of reverse warping the target frame I_{10} back to the reference frame with the LDOF algorithm and our own algorithm. Notice that the LDOF algorithm produces some artifacts in the warped images around the left collar of the shirt, the corner of the lip and the eye. These images show very similar performance for both algorithms which is encouraging since the LDOF approach uses much more sophisticated data and regularization terms in their variational approach.

5 Conclusions and Future Work

We have presented a variational approach to computing dense multi-frame optic flow for non-rigid motion based on a re-parametrisation of the optic flow in terms of a linear combination of a 2D motion basis. The proposed energy formulation reduce number of variables to be computed one order of magnitude to increase computational speed and accuracy of optimisation by applying most generic rank constraint. It is conclusively proven with experiments that we can reduce the problem size and work on global optimization of brightness constancy without actual loss of useful information.

The comparison of our new approach with a state of art optic flow method supports the feasibility of applying statistical rank constraints to the non-rigid registration problem and encourages us to investigate several possible extensions. Future work will include the use of a robust data term which could deal with occlusion and appearance changes by replacing the squared data fidelity term with a more robust one such as the L1 norm. More sophisticated regularisation terms could also be considered such as an anisotropic regulariser.

Acknowledgement. This work is supported by the European Research Council under ERC Starting Grant agreement 204871-HUMANIS. We are grateful to Visesh Chari for sharing his optical flow code and to Adrien Bartoli to share T-shirt sequence.

References

1. Bartoli, A., Gay-Bellile, V., Castellani, U., Peyras, J., Olsen, S., Sayd, P.: Coarse-to-Fine Low-Rank Structure-from-Motion. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, Alaska (2008)
2. Torresani, L., Hertzmann, A., Bregler, C.: Non-rigid structure-from-motion: Estimating shape and motion with hierarchical priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30 (2008)
3. Paladini, M., Del Bue, A., Stosic, M., Dodig, M., Xavier, J., Agapito, L.: Factorization for non-rigid and articulated structure using metric projections. In: IEEE Conference in Computer Vision and Pattern Recognition (2009)
4. Vedula, S., Baker, S., Rander, P., Collins, R.T., Kanade, T.: Three-dimensional scene flow. In: IEEE International Conference of Computer Vision, pp. 722–729 (1999)
5. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proc. International Joint Conference on Artificial Intelligence (1981)
6. Irani, M.: Multi-frame correspondence estimation using subspace constraints. *Int. J. Comput. Vision* 48 (2002)
7. Torresani, L., Yang, D., Alexander, E., Bregler, C.: Tracking and modeling non-rigid objects with rank constraints. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii (2001)
8. Brand, M.: Morphable models from video. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, vol. 2 (2001)
9. Horn, B., Schunck, B.: Determining optical flow. *Artificial Intelligence* 17, 185–203 (1981)
10. Alvarez, L., Weickert, J., Sánchez, J.: Reliable estimation of dense optical flow fields with large displacements. *International Journal of Computer Vision* 39, 41–56 (2000)
11. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3024, pp. 25–36. Springer, Heidelberg (2004)
12. Wedel, A., Pock, T., Braun, J., Franke, U., Cremers, D.: Duality tv-l1 flow with fundamental matrix prior. In: Image and Vision Computing, New Zealand (2008)
13. Becker, F., Wieneke, B., Yuan, J., Schnörr, C.: A variational approach to adaptive correlation for motion estimation in particle image velocimetry. In: Rigoll, G. (ed.) Pattern Recognition 2008. LNCS, vol. 5096, pp. 335–344. Springer, Heidelberg (2008)
14. Wedel, A., Cremers, D., Pock, T., Bischof, H.: Structure- and motion-adaptive regularization for high accuracy optic flow. In: IEEE International Conference on Computer Vision (ICCV), Kyoto, Japan (2009)
15. Bregler, C., Hertzmann, A., Biermann, H.: Recovering non-rigid 3D shape from image streams. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head, South Carolina (2000)
16. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: A factorization approach. *International Journal of Computer Vision* 9 (1992)
17. Torresani, L., Bregler, C.: Space-time tracking. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 801–812. Springer, Heidelberg (2002)

18. Shi, J., Tomasi, C.: Good features to track. In: 1994 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 1994 (1994)
19. Bruhn, A., Weickert, J., Schnörr, C.: Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *International Journal of Computer Vision* 61, 211–231 (2005)
20. Brox, T., Malik, J.: Large displacement optical flow: Descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2010) (to appear)
21. Gelfand, I.M., Fomin, S.V.: *Calculus of Variations*. Dover, New York (2000)
22. Weickert, J., ter Haar Romeny, B.M., Viergever, M.A.: Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Transactions on Image Processing* 7, 398–410 (1998)
23. Varol, A., Salzmann, M., Tola, E., Fua, P.: Template-free monocular reconstruction of deformable surfaces. In: *ICCV 2009* (2009)
24. Weickert, J., Schnörr, C.: Variational optic flow computation with a spatio-temporal smoothness constraint. *Journal of Mathematical Imaging and Vision* 14, 245–255 (2001)

Fast Recovery of Weakly Textured Surfaces from Monocular Image Sequences

Oliver Ruepp and Darius Burschka

Institut für Informatik, Technische Universität München,
Boltzmannstraße 3, D-85748 Garching bei München, Germany
{ruepp,burschka}@in.tum.de

Abstract. We present a method for vision-based recovery of three-dimensional structures through simultaneous model reconstruction and camera position tracking from monocular images. Our approach does not rely on robust feature detecting schemes (such as SIFT, KLT etc.), but works directly on intensity values in the captured images. Thus, it is well-suited for reconstruction of surfaces that exhibit only minimal texture due to partial homogeneity of the surfaces. Our method is based on a well-known optimization technique, which has been implemented in an efficient yet flexible way, in order to achieve high performance while ensuring extensibility.

1 Introduction

Dense recovery of 3D structures from video data is a problem that has been subject to extensive research work, and a number of methods have been developed for dealing with this problem. Some approaches are, e.g., the methods developed by Newcombe et al. [1], Pan et al. [2] and Palaanen et al. [3]. All of those methods rely on presence of salient image features, such as Good Features to Track [4], SIFT [5] features, FAST edges [6] and so on. In some settings, however, the objects do not exhibit much structure, which makes it very hard to find robust, dense feature sets using traditional methods. In such situations, it pays off to use intensity-based methods, which is what we have investigated.

Our method belongs to the family of intensity-based bundle adjustment techniques. An in-depth survey of the original bundle adjustment method is given in the book by Hartley and Zisserman [7]. The paper by Triggs et al. [8] provides a good overview of more recent developments in bundle adjustment and also briefly explains intensity-based approaches. There is also a more recent paper evaluating the status of real-time bundle adjustment methods [9] using sliding window approaches. The main contribution of our work in this context is the combination of sliding-window and intensity-based bundle adjustment.

Basically, the traditional bundle adjustment algorithm facilitates computation of the 3D position of some salient points in a scene from a number of images taken from different viewpoints. The basic idea is as follows: Coordinates of 3D points are associated with features that are recovered from a set of images using feature detection and matching schemes. This approach will obviously work

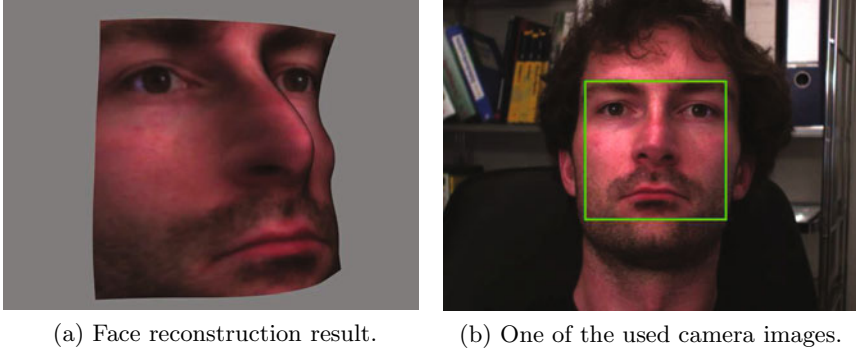


Fig. 1. Example reconstruction result

only if a feature detection scheme is applicable at all. In intensity-based bundle adjustment, we do not assume that robust feature extraction is feasible, and thus we do not work with 2D feature positions, but directly with image intensities.

A number of offline methods for model-based bundle adjustment have been described with applications to face modeling [10, 11]. In contrast to these methods, our method can be used on-line, since the time needed for computing a complete bundle update is comparatively small. For the example image in Figure 1a, the time required to process one frame of the sequence was under 1 second.

In Section 2, we give a detailed explanation of the mathematics involved. This will lead to the formulation of an optimization problem, which we are implementing as described in Section 3. Results have been obtained from real world data sets as well as synthetic data sets, and are presented in Section 4.

2 Mathematical Formulation

There are many possibilities for representing a model of a scene, with the most straightforward one being a point cloud. This is a very general representation that is actually used in the traditional bundle adjustment algorithm, where it works well under the assumption that those points can be reliably identified. As we have mentioned above, we do not assume that this is possible, since we are planning to work exclusively on intensity measurements. Locating a single point in an image simply because of the point's intensity is obviously infeasible, even if several frames are considered simultaneously. This observation disqualifies point clouds as scene representation for intensity based model-recovery algorithms.

2.1 Surface Model

Usually, some additional assumptions need to be made, usually in the form of a specific surface model that ultimately imposes a smoothness constraint on the observed points. Such a model would be a function of type $S : \mathbb{R}^k \times \mathbb{R}^2 \rightarrow \mathbb{R}^3$ that maps a set of k surface parameters together with surface coordinates u, v to

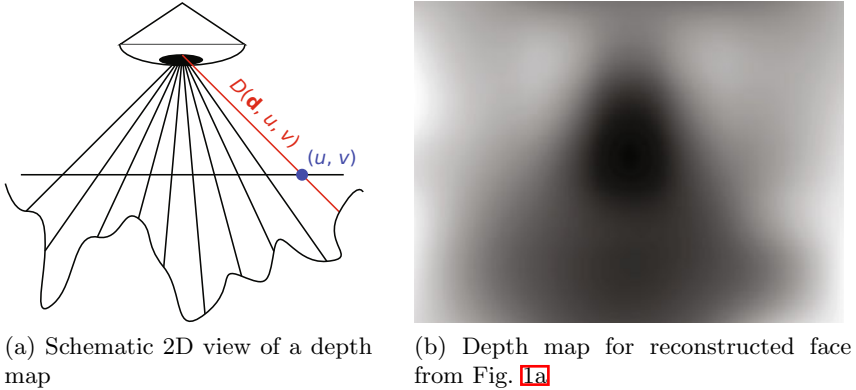


Fig. 2. The depth map concept

three-dimensional spatial coordinates. A model of this type is especially suitable for representation of scenarios that can be described with a small number of parameters k . This loss of generality is a compromise that seems to be necessary in the difficult situation of 3D reconstruction in scenes with low structure.

In this work, we are using surface models that describe the three-dimensional surface indirectly by specifying a per-pixel depth map $D : \mathbb{R}^k \times \mathbb{R}^2 \rightarrow \mathbb{R}$ for a given reference camera image. As illustrated in Figure 2a, the value $D(\mathbf{d}, u, v)$ is supposed to describe the depth for the pixel with coordinates (u, v) . Here, \mathbf{d} is the k -dimensional vector of surface parameters for the depth map.

To put this in a formal mathematical framework, we first need to define some basic characteristics of our camera. As is common, we assume a pinhole model with projection function

$$\pi(\mathbf{p}) = \left(\frac{p_1 f_x}{p_3} + c_x, \frac{p_2 f_y}{p_3} + c_y \right)^T \tag{1}$$

where f_x, f_y are focal lengths in terms of pixel dimensions, c_x, c_y describe the location of the camera center, and $(p_1, p_2, p_3)^T$ is a vector of Cartesian point coordinates. In case of significant radial distortions, the images will be rectified before usage.

Each pixel in the image now corresponds to a ray originating from the camera position that intersects the object surface at a certain depth. Assuming that the camera is located at the origin of the coordinate system, the ray corresponding to pixel coordinates (u, v) can then be parameterized by depth λ , yielding a function $r(u, v, \lambda)$:

$$r(u, v, \lambda) = \lambda \cdot \left(\frac{u - c_x}{f_x}, \frac{v - c_y}{f_y}, 1 \right)^T. \tag{2}$$

We can see now that the composite function $S(\mathbf{d}, u, v) := r(u, v, D(\mathbf{d}, u, v))$ describes a three-dimensional surface. Compared to our general definition of

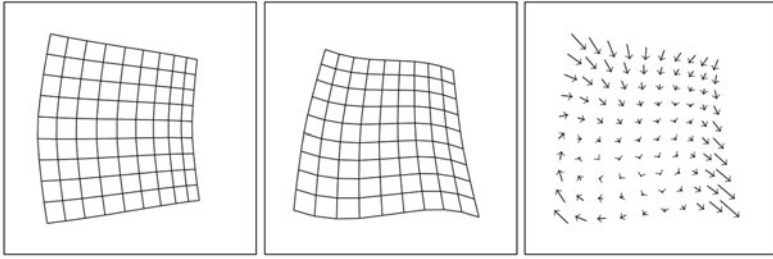


Fig. 3. Left, middle: Surface under two different camera positions. Right: Warping of surface coordinates from left to right image.

a 3D surface as stated earlier, this constitutes a slight restriction. Still, this representation is very well suited to the problem we want to address.

2.2 Optimization Formulation

Observing a static, three-dimensional smooth surface S under two different camera positions will essentially yield two images that are related to each other via a “warping” function. If, for two snapshots of a scene, we exactly know the corresponding extrinsic camera parameters and we have a perfect mathematical description of the surface that we are observing, we can, for each surface pixel in one image, determine the position of that pixel in the other image. In other words, we can formulate a coordinate warping function of type $\mathbb{R}^2 \rightarrow \mathbb{R}^2$ that transforms pixel coordinates from one image to another. Figure 3 shows an example for the coordinate warping function.

Consequently, we would then expect corresponding image intensities to be equal, thus the coordinate warping also defines an image warping, assuming that all pixels are visible. This is the case if there are no occlusions, and the surface does not move out of the camera image. To assure the latter, the depth map we are talking about is not applied to the whole reference frame, but only to a user-chosen rectangular region of interest within the image. As an example, the region of interest is marked with a green rectangle in Figure 1b.

The idea of our approach is now basically the same as in traditional bundle adjustment: Using a nonlinear optimization technique, we are able to compute parameters for the warping function that best explain the observations. Thus, we are able to determine a good approximation of the warping function itself.

To formulate the optimization problem, we need to define a cost function. Before we proceed with the description of that function, we will give a short summary of definitions and notations used. In the following, images are numbered consecutively, and the numbering starts with $n = 0$. Letters set in *italic* represent scalar-valued values, while **bold-faced** letters denote vector-valued quantities.

- \mathbf{d} denotes the k -dimensional vector of parameters describing the depth map.
- $D(\mathbf{d}, u, v)$ denotes the depth map function itself.

- $\mathbf{c}_n = (\mathbf{t}_n, \mathbf{q}_n)$ denotes the extrinsic camera parameters corresponding to image n , consisting of translation vector $\mathbf{t}_n \in \mathbb{R}^3$ and rotation quaternion $\mathbf{q}_n \in \mathbb{R}^4$.
- $T(\mathbf{t}_n, \mathbf{q}_n, \mathbf{p}) : \mathbb{R}^3 \times \mathbb{R}^4 \times \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is a transformation mapping 3D spatial coordinates \mathbf{p} to 3D coordinates in the camera frame described by \mathbf{c}_n .
- $\pi(\mathbf{p})$ is the projection of a 3D point \mathbf{p} to 2D image coordinates.
- $I_n(x, y)$ is the image function of image n . I_0 is the reference image function.

Using this notation, we can define the image coordinate warping function for a certain frame n as follows:

$$w(\mathbf{d}, \mathbf{c}_n, u, v) := \pi(T(\mathbf{c}_n, r(u, v, D(\mathbf{d}, u, v)))). \quad (3)$$

If we knew the perfect model parameters \mathbf{d} and exact camera parameters \mathbf{c}_n for image n , we would expect the relationship $I_n(w(\mathbf{d}, \mathbf{c}_n, u, v)) = I_0(u, v)$ to hold for all model surface coordinates (u, v) .

This leads to the assumption that the correct camera position and the correct model parameters together minimize some difference measure c (e.g., least squares) on intensity values. The optimization process that determines camera and model parameters is computationally quite expensive. Thus, we will not include all possible pixel coordinates in the optimization process, but only the coordinates of m chosen reference points $(u_1, v_1), \dots, (u_m, v_m)$. The corresponding objective function $o(\mathbf{d}, \mathbf{c}_n)$ can then be defined as

$$o(\mathbf{d}, \mathbf{c}_n) = \sum_{i=1}^m c(I_n(w(\mathbf{d}, \mathbf{c}_n, u_i, v_i))) - I_0(u_i, v_i) \quad (4)$$

Our problem of finding a warping function from the template image I_0 to the current image I_n is now formulated as the problem of minimizing the error function with respect to camera and depth map parameters.

We can easily generalize above objective function to a set of views $V = \{\mathcal{V}_1, \mathcal{V}_2, \dots\} \subseteq \{1, 2, \dots, n\}$ (when n images have been acquired) by defining a new objective function

$$o'(\mathbf{d}, \mathbf{c}_{\mathcal{V}_1}, \mathbf{c}_{\mathcal{V}_2}, \dots) = \sum_{j \in V} o(\mathbf{d}, \mathbf{c}_j) \quad (5)$$

This variant is used in our implementation, where we optimize simultaneously over the so-called sliding window of $|V|$ images. In our application, the size $|V|$ of the sliding window is fixed.

Note that above objective functions can easily be modified to formulate a traditional coordinate-based bundle adjustment problem. All we have to do is remove the functions I_j and I_0 from the formula. This is important to know since coordinate-based bundle adjustment will be used to initialize the system.

There are two minor issues that we should also address: Because quaternions are used to represent the rotation of the camera frame, we need to constrain the corresponding parameters \mathbf{q}_n to represent a unit quaternion, and thus, a unit vector. This can trivially be formulated as a constraint $h_1(\mathbf{q}_n) = 0$ with

$h_1(\mathbf{q}_n) = |\mathbf{q}_n|^2 - 1$. Furthermore, it is well-known that reconstruction from monocular images can only be done up to scale. However, it is 0 then at least to enforce a constant scale during the reconstruction process. This can be achieved with the formulation of a constraint $h_2(\mathbf{d}) = 0$ with $h_2(\mathbf{d}) = D(\mathbf{d}, u_1, v_1) - l$ for some constant l .

Since through optimizing above function, we implicitly try to track point positions by intensity values, our approach could have difficulties tracking points in areas with completely homogeneous intensity. Thus, to improve the tracking results, the reference points are chosen from the ROI in such a way that they lie at pixel positions where the image derivative is non-zero.

Furthermore, reference points should be distributed in the region of interest such that the parameters determining the depth map are well constrained. This depends on the specific model used. For a B-Spline depth map model, one will, e.g., need at least a number of reference points that is equal to the number of control points used.

3 Optimization

It is clear that, to actually recover the model parameters from the scene, we need some method to minimize the cost function described above. Since we are dealing with a constrained problem, an adequate method for optimization is Sequential Quadratic Programming (abbreviated as SQP) [12].

The basic idea is as follows: Let $f : \mathbb{R}^k \rightarrow \mathbb{R}$ be a scalar function to be minimized, and let $h : \mathbb{R}^k \rightarrow \mathbb{R}^l$ be a function that describes a constraint of the form $h(x) = 0$ on solutions. It is well-known that for such problems, the so-called Karush-Kuhn-Tucker (KKT) conditions must hold for any value x^* that is a minimum. These conditions can be formulated in equation form as:

$$\begin{pmatrix} \nabla \mathcal{L}(x, \lambda) \\ h(x) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \text{with} \quad \mathcal{L}(x, \lambda) = f(x) + \lambda^T h(x). \quad (6)$$

The term $\lambda \in \mathbb{R}^l$ is the Lagrange multiplier associated with the minimum. This is, in general, a nonlinear system of equations. The Lagrange-Newton Method can be applied to these equations, and we can compute an update Δx to x and a new Lagrange multiplier λ^+ by solving the equation system

$$\begin{pmatrix} \nabla_{xx}^2 \mathcal{L}(x, \lambda) & \nabla_x h(x) \\ \nabla_x h(x)^T & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \lambda^+ \end{pmatrix} = - \begin{pmatrix} \nabla_x f(x) \\ h(x) \end{pmatrix}. \quad (7)$$

When implementing this algorithm, we obviously need to compute the Hessian $\nabla_{xx}^2 \mathcal{L}$ as well as the transposed Jacobian $\nabla_x h$ of h . Since f is, in our case, a quite complex composition of multi-dimensional functions, it is not feasible to compute the exact Hessian. Instead, it is common practice to use the Gauss-Newton approximation of the Hessian, as detailed below.

Note that the Hessian of $\lambda^T h(x)$ is, on the other hand, easy to determine: Since our constraints are all linear or quadratic, all second-order derivatives are reasonably easy to compute, so the exact Hessian can be established.

3.1 Implementation Details

In our case, the objective function f is the composition $c \circ g$ of a scalar cost function c with some multi-dimensional comparison function g . The typical choice for a cost function would be the least-squares cost $g(x) = x^T x$, but it is well-known that this cost function is very susceptible to outliers. Thus, we are instead using the pseudo-Huber cost function [7, p. 619], which is known to be very robust. Independent of the actual cost function used, the Hessian of the total cost is approximated as:

$$\nabla_{xx}^2(c \circ g)(x) \approx (\nabla_x g)(x) \cdot (\nabla_{xx}^2 c)(f(x)) \cdot (\nabla_x g)^T(x)$$

We see that in order to estimate the Hessian, we need to compute the Jacobian of the comparison function g . That Jacobian is sparse, which means that the Hessian will also be sparse. As in the traditional bundle adjustment algorithm, exploiting the sparsity structure is extremely important.

The sparse Jacobians of f and h are computed using our own variant of Automatic Differentiation (AD) [13]. Our method can be described as a hybrid between traditional AD and symbolic differentiation. Traditional AD basically treats all functions as function compositions. AD is able to compute derivatives of elementary functions, such as basic arithmetic, cos, sin, exp and so on, directly, while the derivatives of composite functions are computed according to the chain rule.

The accuracy of AD is very good, especially when compared to finite difference approximation. Derivatives computed by AD are usually accurate up to machine precision. Additionally, the performance of AD is known to be good as well, since the time required to evaluate derivatives of a function is proportional to the time needed to evaluate the original function. However, when evaluating sparse Jacobians, plain AD is usually not optimal, and finding the most efficient way to compute sparse Jacobians within the framework of AD is actually an NP-complete problem [14]. That problem is solved in state-of-the-art AD implementations by means of heuristics.

Our approach for computing sparse Jacobians also relies on the chain rule, but it is applied at a different level: For a composition of vector-valued functions $f_1 \circ f_2 \circ \dots \circ f_n$, the Jacobian can be computed as a sparse chained matrix product $J_{f_1} \cdot J_{f_2} \cdot \dots \cdot J_{f_n}$. We choose to implement computation routines for the Jacobians J_{f_i} directly instead of starting with differentiating the most basic functions. The overall Jacobian is then evaluated as sparse matrix product with optimized bracketing. It is well-known that the bracketing of a matrix chain product is essential to evaluation performance [15, p. 331]. This observation also applies to sparse chain matrices, and is also the basis of a highly-efficient heuristic for traditional AD [16]. In our case, the sparsity structure of the Jacobians never changes, so the optimal bracketing that has been determined once for a certain objective function remains the same.

Implementing Jacobian evaluation code for vector-valued functions directly obviously involves additional work and is error prone, but our method also has two advantages over traditional AD:

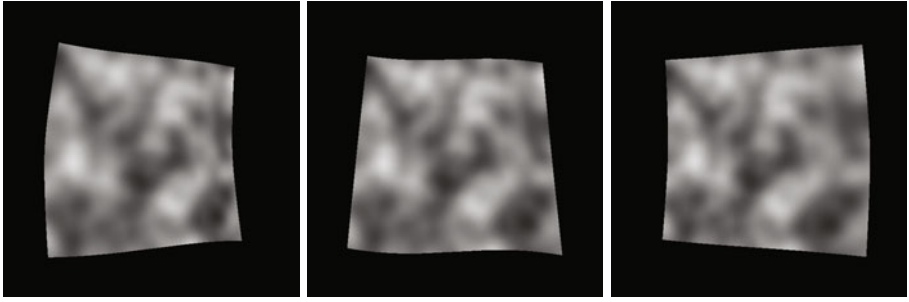


Fig. 4. Samples from artificial image sequence

- Speed: When used properly, our method can be substantially faster than traditional AD. For the coordinate transformation function T , as described in Section 2, we have compared the performance of our own implementation and that of an ADOL-C [17] based variant, and found that our method is about three times as fast.
- Flexibility: Some functions resist treatment by AD, especially in cases where the function to be differentiated is defined in a piecewise fashion. This is the case for our image functions, which are made continuous and differentiable using bicubic interpolation. However, it is reasonably simple for a human to implement efficient evaluation algorithms for the interpolated image function as well as its derivatives.

After computation of the Jacobian is finished, the approximate Hessian can be evaluated and the QP system is solved repeatedly. For increased robustness of this process, we add a damping term λI to the Hessian of \mathcal{L} . This method is well-known in the context of Levenberg-Marquardt optimization [18, 19] and can be applied to the SQP method as well. The equation system itself is then solved by employing a sparse Cholesky transformation on the whole system. The efficient Eigen library for Linear Algebra¹ is used to handle this. Finally, when a solution to the system has been found, a simple step size search according to the Armijo rule is performed.

3.2 Bundle Adjustment

Now that we have established the tools to solve the original optimization problems stated in Section 2, we can describe how the methods are actually applied to achieve model reconstruction.

First of all, we need to initialize the system. We assume that the user has chosen the template image and an ROI for the surface to be reconstructed. For a given window size $|V|$, we acquire images $\{1, \dots, |V|\}$ and compute the optical flow [20] on these, which yields initial correspondences between the template image and the images to be used for initialization. One run of full coordinate-based bundle adjustment on those correspondences yields a starting point for further

¹ <http://eigen.tuxfamily.org/>

intensity-based optimization, which is carried out after the initialization through optical flow. While the coordinate-based optimization serves to provide a rough estimate of the surface and camera positions, the intensity based optimization is used to refine the initially found parameters.

After the initialization, the algorithm alternates between coordinate based optimization steps (optimizing towards optical flow results) and intensity based optimization. No full bundle adjustment on optical flow information is performed any more. Instead, the coordinate based optimization is used only to determine the camera position. With camera parameters initialized, a full intensity-based bundle optimization is carried out to refine the parameters.

To achieve usable results and assure that the algorithm is stable, one must also be careful which images to select for the optimization window V . Simply choosing consecutive images might be a bad idea, because if the camera stops moving for some time, a number of images from the same position will be taken. If those images are placed inside the window, the optimization process will become very unstable, since depth reconstruction is impossible without a certain minimal baseline.

In our implementation, new images in the window are only accepted if their difference of baseline to the previous image in the window is big enough. Whenever a new image is accepted, the oldest image in the window is discarded. This is obviously a rather crude algorithm, but it helps to avoid the most obvious problems. We are planning to implement more sophisticated methods in the future.

4 Results

We have tested our algorithm on a set of artificial rendered image sequences, as well as on sequences of real scenes. The artificial data set was useful for generating images with known ground truth, while the sequences of real images have been used to show that the approach also works in the “real world.” As depth map model, we have used B-Spline surfaces of varying order and complexity.

Our first tests were on artificial images generated by a renderer. Here, we show results for one of the used sequences. Figure 4 shows an example image from the sequence, showing a surface with a texture that exhibits only intensity gradients, and almost no structure. Because we wanted to get a rough idea of how well traditional approaches would work on that sequence, we ran a SIFT feature detector on some of the images. The feature detection process resulted in about 20 features, depending on the actual image. Even when assuming that all features can be reliably identified through the whole sequence, and that no false feature matchings occur, this is by far not enough to fully describe the complexity of the actual surface. The surface is a quadratic spline surface determined by 25 control points (5 in each direction).

To compare the reconstructed surface to the ground truth, we have determined the normalized cross-correlation (NCC) between the ground truth surface of the sequence and the reconstruction result. The result of this comparison was that the reconstruction is extremely accurate, yielding surface models that achieved a NCC ratio of over 0.96, where 1 is the best possible value.



Fig. 5. Samples from real-world image sequence



Fig. 6. Template image, depth map, and 3D rendering

The artificial sequences have been used because it is really difficult in a real-world scenario to determine the ground truth. Still, it is important to show that our approach also works on actual data generated from a camera. Hence, we have tested our method on a scene that was showing a piece of white cloth draped over a cup. You can see some images of the recorded sequence in Figure 5. Figure 6 shows the template image, the associated depth map, and the resulting 3D model.

As can be seen, the reconstruction quality is still good, despite the lack of texture in the scene. It is clearly possible to recognize the shape of the cup underneath the cloth.

As for running times: Our algorithm has been tried on a system with an Intel Core i7-820QM 1.73 GHz quad core CPU, using only one of the CPU cores. Running time generally depends on the resolution of the surface model, but generally, one frame was processed in under one second. The major time spent during reconstruction was due to intensity-based optimization. The convergence of the intensity-based optimization was rather slow, which is probably due to the non-convex nature of the cost function in case of large displacements of the tracked pixels to the optimal position. Still, the performance is promising, and we expect it to be possible to further improve performance by pursuing more elaborate optimization schemes.

5 Conclusion

The basis for further research has been established with our monocular model recovery and tracking algorithm. There are many possible extensions and improvements to this technique.

First of all, while the reference-point based reconstruction works well, it would probably constitute a major improvement if we were able to capture, in addition to point intensity values, a comparison of texture gradients in the area surrounding the reference points. We would expect this to further improve the stability and convergence speed of the optimization method.

While the algorithm is already quite fast, there is still a lot of potential for speed improvement. There exist more specialized algorithms that could be used for solving the QP equations [21]. GPU algorithms could be used for performing image subsampling, which constitutes the major part of the current computation time consumption. Finally, it should also be possible to speed up the involved geometry computations using the GPU.

Furthermore, we did not address the issue of changing illumination conditions. We would like to be able to deal with changes in brightness, but also with specularities, which would, in the current approach, both cause severe problems. However, some techniques for dealing with problems of that kind have already been developed, e.g., normalized cross-correlation matching for brightness-invariant matching. It should be possible to integrate them into our method.

We would also like to extend the approach such that deformable surfaces can be reconstructed and tracked. For tackling this problem, we intend to use a setup of two independently moving cameras. Based on such an idea, we would like to introduce a method for determining deformation parameters, allowing us also to predict and simulate deformations. We see applications for such a technique mainly in medical imaging.

Acknowledgement. This research has been funded by The German Heart Centre Munich.

References

1. Newcombe, R., Davison, A.: Live dense reconstruction with a single moving camera. In: IEEE Conference on Computer Vision and Pattern Recognition CVPR (2010)
2. Pan, Q., Reitmayr, G., Drummond, T.: ProFORMA: Probabilistic Feature-based On-line Rapid Model Acquisition. In: Proc. 20th British Machine Vision Conference (BMVC), London (2009)
3. Paalanen, P., Kyrki, V., Kamarainen, J.K.: Towards Monocular On-Line 3D Reconstruction. In: Workshop on Vision in Action: Efficient strategies for cognitive agents in complex environments, Marseille France, Markus Vincze and Danica Kragic and Darius Burschka and Antonis Argyros (2008)
4. Shi, J., Tomasi, C.: Good features to track. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 1994), pp. 593–600 (1994)
5. Lowe, D.G.: Object recognition from local scale-invariant features. In: ICCV, pp. 1150–1157 (1999)
6. Rosten, E., Drummond, T.W.: Machine learning for high-speed corner detection. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 430–443. Springer, Heidelberg (2006)
7. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004) ISBN: 0521540518

8. 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)
9. Engels, C., Stewénius, H., Nistér, D.: Bundle adjustment rules. In: Photogrammetric Computer Vision (PCV), ISPRS (2006)
10. Fua, P.: Using model-driven bundle-adjustment to model heads from raw video sequences. In: Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 1, pp. 46–53 (1999)
11. Shan, Y., Liu, Z., Zhang, Z.: Model-based bundle adjustment with application to face modeling. In: Proceedings of Eighth IEEE International Conference on Computer Vision, ICCV 2001, vol. 2, pp. 644–651 (2001)
12. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, Heidelberg (2000)
13. Griewank, A., Walther, A.: Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation, 2nd edn. Other Titles in Applied Mathematics, vol. 105. SIAM, Philadelphia (2008)
14. Naumann, U.: Optimal jacobian accumulation is np-complete. *Math. Program.* 112, 427–441 (2007)
15. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 2nd edn. The MIT Press and McGraw-Hill Book Company (2001)
16. Griewank, A., Naumann, U.: Accumulating jacobians as chained sparse matrix products. *Math. Program.* 95, 555–571 (2003)
17. Griewank, A., Juedes, D., Utke, J.: Algorithm 755. ADOL-C: A package for the automatic differentiation of algorithms written in C/C++ 22, 131–167 (1996)
18. Levenberg, K.: A method for the solution of certain non-linear problems in least squares. *Quarterly Journal of Applied Mathematics* II, 164–168 (1944)
19. Marquardt, D.W.: An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics* 11, 431–441 (1963)
20. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision (darpa). In: Proceedings of the 1981 DARPA Image Understanding Workshop, pp. 121–130 (1981)
21. Davis, T.A., Hager, W.W.: Dynamic supernodes in sparse cholesky update/downdate and triangular solves. *ACM Trans. Math. Softw.* 35, 1–23 (2009)

Ghost-Free High Dynamic Range Imaging

Yong Seok Heo¹, Kyoung Mu Lee¹, Sang Uk Lee¹,
Youngsu Moon², and Joonhyuk Cha²

¹ Department of EECS, ASRI, Seoul National University, Seoul, Korea

<http://cv.snu.ac.kr>

² Samsung Advanced Institute of Technology, Samsung Electronics Co.,
Yong-In, Korea

Abstract. Most high dynamic range image (HDRI) algorithms assume stationary scene for registering multiple images which are taken under different exposure settings. In practice, however, there can be some global or local movements between images caused by either camera or object motions. This situation usually causes ghost artifacts which make the same object appear multiple times in the resultant HDRI. To solve this problem, most conventional algorithms conduct ghost detection procedures followed by ghost region filling with the estimated radiance values. However, usually these methods largely depend on the accuracy of the ghost detection results, and thus often suffer from color artifacts around the ghost regions. In this paper, we propose a new robust ghost-free HDRI generation algorithm that does not require accurate ghost detection and not suffer from the color artifact problem. To deal with the ghost problem, our algorithm utilizes the global intensity transfer functions obtained from joint probability density functions (pdfs) between different exposure images. Then, to estimate reliable radiance values, we employ a generalized weighted filtering technique using the global intensity transfer functions. Experimental results show that our method produces the state-of-the-art performance in generating ghost-free HDR images.

1 Introduction

Typical cameras represent a pixel using only 256 values for each of the red, green, and blue channel. On the contrary, the range of radiance of a real scene has a far wider range than 256 values [1]. Hence, a photograph taken by a conventional camera can not capture the whole dynamic range of scene radiance. So, the cameras usually compress the scene radiance value using a proper function which is often called the camera response function (CRF). This process, however, can cause unpleasant under- or over-exposed regions.

Many approaches have been proposed to recover a high dynamic range image (HDRI) by estimating the CRF using multiple low dynamic range images (LDRI) which are taken under different exposure settings for the scene [2-5].

The pioneering work of Mann and Picard [2] used a gamma function to estimate a CRF. Debevec and Malik [3] estimated a CRF using error function with smoothness constraint in a least squared-error sense, and then the radiance value of each pixel is determined by a weighted sum of the radiance values of multiple exposure images. Mitsunaga and Nayar [4] approximated a CRF using a polynomial with a fixed degree. They only assumed that the ratios of the exposures between images are roughly known, instead of the exposure time. Grossberg and Nayar [5] suggested a robust method to recover a CRF using intensity histograms instead of a pixel value itself without image registration.

In practice, however, while fusing multiple images into a single radiance image, all these methods severely suffer from artifacts caused by moving camera and/or objects, because they assume a stationary scene. A camera motion causes global image transformation such as an affine or perspective transformations between different exposure images. If one takes photographs using a tripod, this problem might be reduced. A more critical problem, however, is caused by an object motion which invokes inevitable ghost artifacts that make the same object appears multiple times in a generated HDRI. Due to these reasons, practically it is a very important and critical issue to produce a ghost-free HDRI from multiple images.

In this paper, we propose a new HDRI generation method that is very effective for handling global and local movements from multiple exposures. Fig. 1 shows an example of our HDRI generated from multiple LDRI with object motions. Compared with the standard method [3], our proposed method produces much clear and ghost-free HDRI.



Fig. 1. (a)-(d) are input LDRI. (e) Result of the standard HDRI method [3]. (f) Result of the proposed method.

2 Previous Works

There are several works for handling ghost artifacts for HDRI generation. Jacobs *et al.* [6] compared two measures to detect ghost regions such as variance image (VI) [1] and uncertainty image (UI) [6]. They argued that VI was effective for detecting high contrast movement such as moving people, cars, and etc., while UI was effective for low contrast movement such as moving leaves and water rippling [6]. Grosch [7] detected ghost regions using predicted pixel colors which were estimated from the CRF. They defined an error map that had invalid pixel set by thresholding the absolute difference value between the predicted pixel color and the original color. However, These methods have a common drawback that the ghost detection results tend to be sensitive to the threshold values of those measures. Also, they can decrease the dynamic range of ghost regions, since they fill the detected ghost regions with the radiance values from only a single image.

Some other approaches utilize as many multiple exposures as possible for ghost regions. Gallo *et al.* [8] detected ghost pixels using a linear property of log radiance values in block-wise comparison. For each pixel, they combined multiple exposures except for the images that had ghost regions. Then, they blended the block boundaries to reduce the color difference between neighboring blocks. Raman *et al.* [9] suggested a similar approach. They also detected ghost regions using block-based comparison between different exposures followed by thresholding. Then, they performed the Poisson blending between neighboring blocks. However, these methods still suffer from color artifacts around block boundaries due to inaccurate CRF estimation [8].

Alternatively, there are other approaches that solve this problem by adjusting weighting function in the Debevec and Malik's weighted average framework [3]. Khan *et al.* [10] suggested a ghost removal method by adjusting weights when combining multiple exposures. They assumed that all pixels were belonging to either foreground (moving part) or background (static part), and those background pixels were significantly prevailed than the pixels of foreground moving objects. Their weight function is composed of two terms regarding the probability of being correctly exposed and the probability of belonging to the background. Then, they iteratively updated the probability of belonging to the background. Pedone and Heikkilä [11] suggested a similar approach to [10]. They estimated bandwidth matrices for computing the accurate probability of belonging to the background, and propagated the influence of the low probabilities to the surrounding regions using an energy minimization technique. A main drawback of these methods is that if the object of interest is moving, that object can be recognized as ghost and disappeared in the resultant HDRI.

On the other hand, Bogoni [12] estimated motion vectors using optical flow for different exposure images, and then used those information to warp other exposure images. Kang *et al.* [13] also used a gradient-based optical flow method to find corresponding pixels between neighboring images that had alternating different exposures for producing HDR video sequences. However, it is not trivial to find the accurate correspondence between different exposure images.

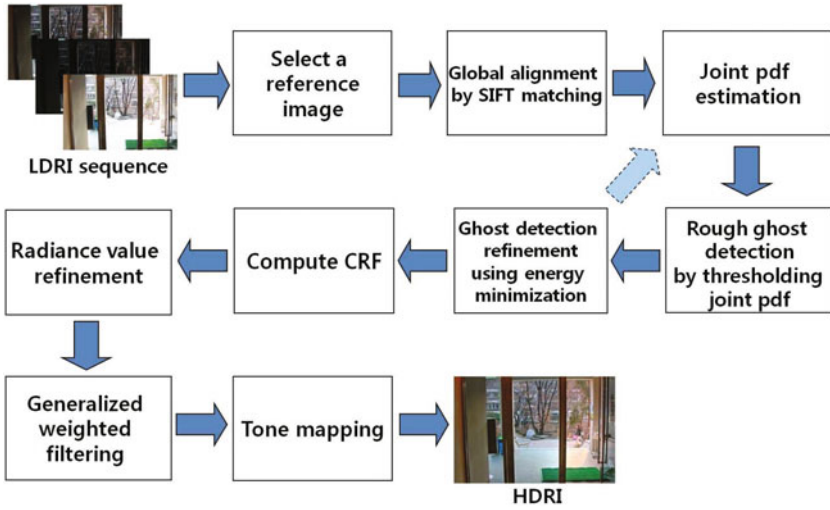


Fig. 2. An overview of our approach

Most approaches described above determine whether a pixel in each image is a ghost or not in deterministic or statistical manners. Then, to compute the radiance values, they utilize the un-ghost pixels of multiple exposures only in the same position. In this case, when ghost detection is not so accurate enough, many artifacts arise in the resultant HDRI. Also, even when the ghost detection is acceptable, it is still problematic to fill those regions with proper radiance values. Filling those regions using only a single exposure image can reduce the dynamic range, and also filling them using only un-ghost pixels can produce color artifacts around the ghost regions due to the inaccurate CRF estimation.

3 Proposed Algorithm

Fig. 2 shows an overview of our algorithm. First, we select a reference image to generate a HDRI among multiple exposure images. Then, we globally align other images to this reference image to handle camera motions caused by handshakes. Next, we estimate the joint probability density functions (pdfs) between the reference image and other images to estimate the global intensity transfer functions. Based on these joint pdfs, we roughly detect ghost regions in other images w.r.t. the reference image. Then, a refinement procedure is followed based on a global energy minimization framework using Graph-cuts [14]. The joint pdf and ghost detection processes are performed recursively for two or three steps. After that, we compute a CRF by sampling some un-ghost pixels. Based on this CRF, we refine the radiance values of other images w.r.t. that of the reference image to reduce the CRF estimation error. Finally, using the refined radiance

values and the global intensity transfer functions of all exposure images, we perform a generalized weighted filtering to compute the final radiance values. After tone-mapping process, we produce a ghost-free HDRI for the reference image. Detailed explanation is as follows.

3.1 Reference Image Selection and Global Image Alignment

First, we have to select a reference image among multiple exposure images. We choose the image that has least saturated regions such as under- or over-exposed regions as the reference image. Then, we globally align other images to this reference image. In this global alignment, we use SIFT feature-based alignment method [15], since SIFT descriptor is robust to exposure changes [15] and it can handle affine or perspective image transformations in some degree. After finding SIFT features, we compute homographies based on these features using RANSAC [16], and then warp other images to the reference image. However, even after this global alignment, there still remain some local misalignments due to moving objects that usually cause ghost artifacts. Let us describe how to deal with this problem in detail in the following sections.

3.2 Joint Pdf Estimation

To deal with ghost artifacts, we need a measure to judge the correspondence between different exposure images. For this measure, we use a global intensity relationship [17]. To estimate the global relationship between different exposures, we construct a joint histogram $P_{n_0, n}^k$ for each color channel $k \in \{R, G, B\}$ between the reference n_0^{th} image and other n^{th} image. $P_{n_0, n}^k$ is defined by

$$P_{n_0, n}^k(i, j) = \frac{1}{M} \sum_p G_n(p) \cdot T[(i, j) = (I_{n_0}^k(p), I_n^k(p))], \quad (1)$$

where p is a pixel position index. $T[\cdot]$ is one if the argument is true, zero otherwise. M is the total number of corresponding pixels in an image. $I_n^k(\cdot)$ is an intensity value of k channel of n^{th} image. $G_n(\cdot)$ is a ghost weight function which is defined in the following section. At first iteration, we set $G_n(\cdot) = 1$ for all pixels of all exposures. Next, Parzen windowing is performed by convolving 2D Gaussian function to have smooth joint pdfs. In this work, we used a 5×5 Gaussian function. Then, we normalize the joint histograms such that the sum of all the elements equals to one. Examples of joint pdfs are shown in Fig. 3 (d)-(f) for the images in Fig. 3 (a)-(c), where (a) is the reference image.

3.3 Ghost Region Estimation

For each n^{th} image except for the reference image, we detect ghost pixels by defining ghost weight $G_n(\cdot)$ which is defined by

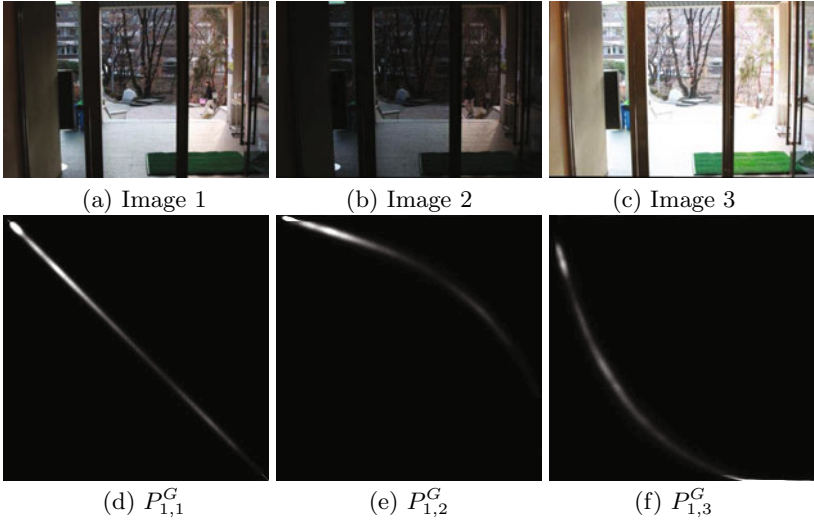


Fig. 3. (a)-(c) are input image sequence, where (a) is the reference image. (d)-(f) are joint pdfs of the green channel corresponding to (a)-(c), respectively.

$$G_n(p) = \begin{cases} 0, & \text{if } P_{n_0,n}^R(I_{n_0}^R(p), I_n^R(p)) < c \text{ or} \\ & P_{n_0,n}^G(I_{n_0}^G(p), I_n^G(p)) < c \text{ or} \\ & P_{n_0,n}^B(I_{n_0}^B(p), I_n^B(p)) < c \\ 1, & \text{otherwise} \end{cases} \quad (2)$$

$G_n(p) = 0$ represents that a pixel p in the n^{th} image is a ghost, while $G_n(p) = 1$ represents that the pixel p is a non-ghost pixel. For the reference image, all the pixels are assumed to be non-ghost pixels. In this work, we set the threshold c as 10^{-5} . These ghost regions initially determined by thresholding joint pdfs could be very noisy and inaccurate. Hence, we refine the ghost detection result by using an energy minimization approach. For each image, we define the total energy to minimize as follows:

$$E(f_n) = \sum_p D_p(f_n(p)) + \sum_p \sum_{q \in N(p)} V_{pq}(f_n(p), f_n(q)), \quad (3)$$

where the Boolean label $f_n \in \{0, 1\}$ represents whether a pixel is a ghost or not. When $f_n(p) = 0$, a pixel p in the n^{th} image is a ghost, while $f_n(p) = 1$ represents that a pixel p in the n^{th} image is not a ghost pixel. $N(p)$ represents a neighboring pixels of p . In this work, we use a four-neighborhood system. Our data cost $D_p(\cdot)$ is defined by

$$D_p(f_n(p)) = \begin{cases} 0, & \text{if } (f_n(p) = 0 \wedge G_n(p) = 0) \text{ or} \\ & (f_n(p) = 1 \wedge G_n(p) = 1) \\ \beta, & \text{otherwise} \end{cases}, \quad (4)$$

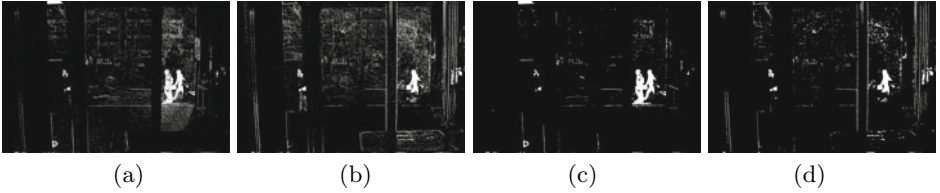


Fig. 4. Ghost detection result. Image 1 (in Fig. 3 (a)) is the reference image. (a) and (b) are the ghost regions corresponding to image 2 and image 3, respectively, using thresholding joint pdf. (c) and (d) are the refined ghost regions corresponding to image 2 and image 3, respectively, using global energy minimization.

where β is a constant, which we set as $\beta = 2.5$. We define a smoothness cost $V_{pq}(\cdot, \cdot)$ as a truncated linear function defined by

$$V_{pq}(f_n(p), f_n(q)) = \lambda_{pq} \cdot \min(|f_n(p) - f_n(q)|, V_{\max}),$$

$$\lambda_{pq} = \begin{cases} \lambda_L, & \text{if } \{(|I_{n_0}(p) - I_{n_0}(q)| < \eta) \vee \\ & (|I_n(p) - I_n(q)| < \eta)\} \\ \lambda_S, & \text{otherwise} \end{cases}, \quad (5)$$

where $\lambda_L > \lambda_S$, and $I_n(p)$ represents a gray value of a pixel p in the n^{th} image. Note that the strength of $V_{pq}(\cdot, \cdot)$ depends on the intensity difference. If the intensity change between neighboring pixels is smaller than the threshold η , we emphasize more smoothness by choosing a larger λ_L value than a smaller one, λ_S . In this work, we set variables in Eq. (5) as follows: $V_{\max} = 1$, $\eta = 5$, $\lambda_L = 3.0$, $\lambda_S = 1.0$. The total energy $E(f_n)$ is optimized using the Graph-cuts (alpha-expansion) algorithm [14]. Using optimized $f_n(\cdot)$, $G_n(\cdot)$ is also updated. Estimating joint pdf and ghost detection processes are iteratively updated. Empirically, two or three iterations are sufficient for convergence.

A ghost estimation example for Fig. 3 (a)-(c) is shown in Fig. 4, where white pixels represent the ghost pixels. We can clearly see that, after global energy minimization, ghost detection results become less noisy and more accurate than those of naive thresholding. Also, it is worth noting that our method does not directly use these ghost detection results, since these results can be still erroneous. Instead, we apply more robust filtering method, which is described in section 3.5.

3.4 Camera Response Function Estimation and Radiance Value Refinement

If we assume that exposure time Δt_n is known, the radiance value of a pixel p in the n^{th} image can be obtained [3] by

$$\ln E_n^k(p) = g(I_n^k(p)) - \ln \Delta t_n, \quad (6)$$

where E represents a radiance value and $g(\cdot)$ is an inverse camera response function. Note that our actual goal is to compute the radiance values for all the pixels in the reference image. Hence, to estimate the radiance value, we should estimate the inverse CRF function $g(\cdot)$.

To compute $g(\cdot)$, we randomly sample a number of points (55 in this work), avoiding ghost regions and edge regions. Then, we computed $g(\cdot)$ using the method [3]. Also, as [3] suggested, combined radiance value is computed as a weighted average as follows:

$$\ln E^k(p) = \frac{\sum_n w(I_n^k(p))(g(I_n^k(p) - \ln \Delta t_n))}{\sum_n w(I_n^k(p))}, \quad (7)$$

where $w(\cdot)$ is a triangle-shaped function defined by

$$w(z) = \begin{cases} z - z_{\min} & \text{for } z \leq 0.5(z_{\min} + z_{\max}) \\ z_{\max} - z & \text{for } z > 0.5(z_{\min} + z_{\max}) \end{cases}, \quad (8)$$

where z_{\min} and z_{\max} are the minimum and maximum intensity values, respectively.

To eliminate ghost artifacts, we should combine a set of exposure images which does not include ghost pixels in calculating Eq. (7). However, even if we accurately detect ghost pixels, there can be still significant color artifacts due to inaccurate CRF estimation [8]. In other words, averaging from different sets of exposure images which do not include ghost pixels often induces significant color differences, because each E value of the same position for different exposure images could have different values owing to inaccurate $g(\cdot)$.

The estimated CRF can be inaccurate by various factors such as inaccurate ghost detection, image alignment error, noise and blurring. To solve this problem, we refine the radiance values of other images such that all the pixels of different exposure images have consistent radiance values to the reference image. First, for non-ghost pixels, we compute the radiance values using Eq. (7). Then, we obtain refined radiance value $\bar{E}_n^k(\cdot)$ by averaging the radiance values for each exposure images as follows:

$$\ln \bar{E}_n^k(z) = \frac{1}{C} \sum_{p, I_n^k(p)=z} G_n(p) \cdot \ln E^k(p), \quad (9)$$

where C is a normalization constant. To acquire more smooth curve, we can adopt a more sophisticated curve-fitting algorithm [4].

3.5 Generalized Weighted Filtering Method

In this section, we propose a robust weighted filtering approach for HDRI generation. Fig. 5 depicts our generalized weighted filtering scheme. First, using the estimated joint pdf, the global intensity transfer function between the reference n_0^{th} image and the n^{th} image can be computed in the minimum mean squared error (MMSE) sense. To consider the saturated cases, we define the global intensity transfer functions according to the exposure time of images as follows:

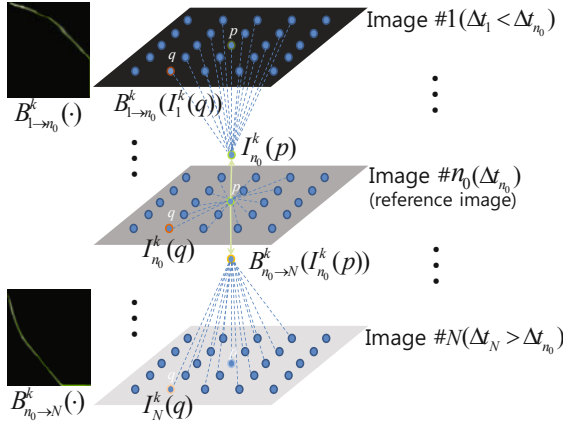


Fig. 5. Generalized weighted filtering scheme. The number of input images is N . Input images are aligned according to exposure time. The final radiance value \hat{E} of the reference n_0^{th} image is computed using weighted sum of refined radiance values \bar{E}_n of images. For each pixel q of n^{th} image in the window $L(p)$ centered at pixel p , the total weight is determined by combining three weights; properly exposed weight $w(\cdot)$, geometric distance weight $d(p, q)$, and color difference weight $c_{n_0, n}(p, q)$.

$$\begin{aligned}
 B_{n_0 \rightarrow n}^k(i) &= \frac{\sum_{j=0}^{255} P_{n_0, n}^k(i, j) \cdot j}{\sum_{j=0}^{255} P_{n_0, n}^k(i, j)} \quad (\text{for } \Delta t_{n_0} < \Delta t_n), \\
 B_{n \rightarrow n_0}^k(j) &= \frac{\sum_{i=0}^{255} P_{n_0, n}^k(i, j) \cdot i}{\sum_{i=0}^{255} P_{n_0, n}^k(i, j)} \quad (\text{for } \Delta t_{n_0} \geq \Delta t_n).
 \end{aligned}
 \tag{10}$$

For a pixel p in the reference image, we define a window region $L(p)$ that includes all the pixels around the center pixel p in all the exposure images. Then, to compute the final radiance value $\hat{E}^k(\cdot)$, we compute a weighted sum based on the bilateral filtering weight [18] and the intensity weight as follows:

$$\begin{aligned}
 \ln \hat{E}^k(p) &= \frac{\sum_n \sum_{q \in L(p)} w(I_n^k(q)) c_{n_0, n}(p, q) d(p, q) \ln \bar{E}_n^k(I_n^k(q))}{\sum_n \sum_{q \in L(p)} w(I_n^k(q)) c_{n_0, n}(p, q) d(p, q)}, \\
 c_{n_0, n}(p, q) &= \exp\left(\frac{-\sum_k \psi_{n_0, n}^k(p, q)}{\sigma_c^2}\right), \\
 \psi_{n_0, n}^k(p, q) &= \begin{cases} |B_{n_0 \rightarrow n}^k(I_{n_0}^k(p)) - I_n^k(q)|^2 & (\text{for } \Delta t_{n_0} < \Delta t_n) \\ |I_{n_0}^k(p) - B_{n \rightarrow n_0}^k(I_n^k(q))|^2 & (\text{for } \Delta t_{n_0} \geq \Delta t_n) \end{cases}, \\
 d(p, q) &= \exp\left(\frac{-\|p - q\|^2}{\sigma_d^2}\right),
 \end{aligned}
 \tag{11}$$

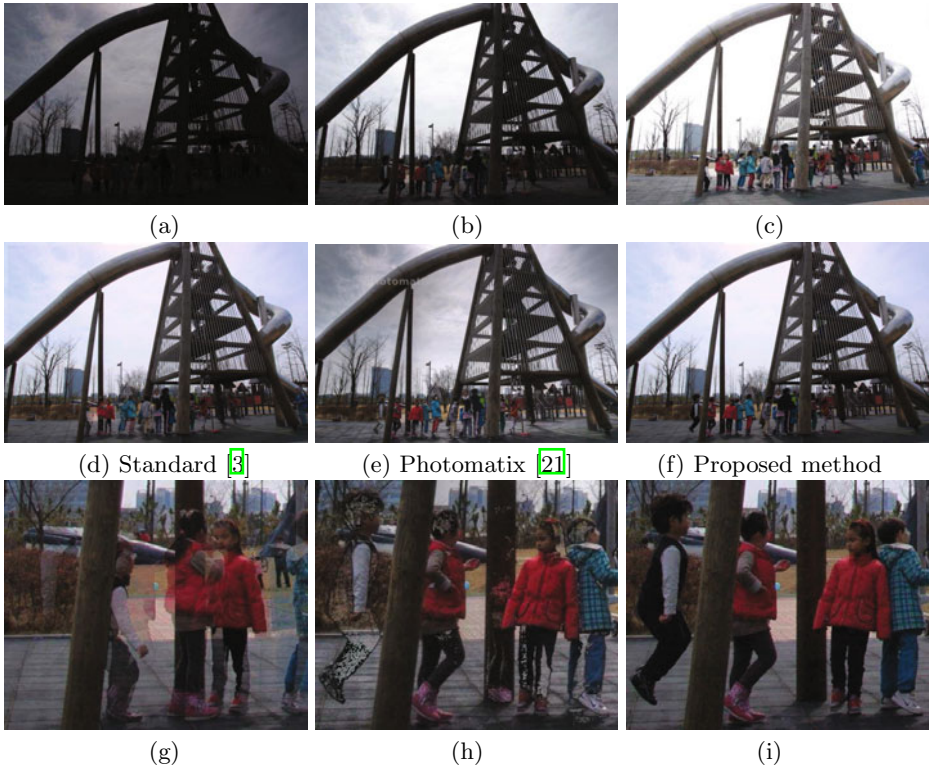


Fig. 6. Playground sequence. (a)-(c) are input LDR images, where (b) is the reference image. (d) Result of the standard method [3]. (e) Result of the Photomatix [21]. (f) Result of the proposed method. (g)-(i) are the magnified views of (d)-(f), respectively.

where $w(\cdot)$ is defined in Eq. (8) that emphasizes properly exposed intensity. $c_{n_0, n}(\cdot, \cdot)$ is a weighting function for color difference between two pixels, $d(\cdot, \cdot)$ is a weighting function for geometric distance between two pixels, and $\|\cdot\|$ represents the Euclidean distance. Note that, in Eq. (11), we use the refined radiance values $\bar{E}_n^k(\cdot)$ in Eq. (9) instead of the radiance values from the estimated CRF function, because the refined radiance values produce less color artifacts when combining radiance values of multiple exposures. Also, in order to compute $c_{n_0, n}(\cdot, \cdot)$, using the global intensity transfer functions in Eq. (10) the intensity of the reference n_0^{th} image is transformed to n^{th} image that has longer exposure time than that of the n_0^{th} image. Conversely, for n^{th} image that has shorter exposure time than that of n_0^{th} image, the intensity of n^{th} image is transformed to the n_0^{th} image. This weighted filtering approach can be considered as a generalized Debevec & Malik approach [3] in that it considers a wider range of pixels, and is robust to ghost artifacts, image misalignments and CRF estimation error. In this work, we set variables in Eq. (11) as $\sigma_c = 7$, $\sigma_d = 10$. The size of $L(p)$ was set as 21×21 for each exposure image.

4 Experimental Results

To evaluate the performance of our method, we tested our algorithm for various scenes that include camera and object movements. To visualize computed radiance values, we averaged the results of both gradient-based [19] and photographic [20] tone-mapping methods.

First, we compared our method with the commercial Photomatix software [21]. For Photomatix software, we tried to reduce ghost artifacts with ‘moving objects/people’ and ‘high’ detection modes. Fig. 6 and 7 (a)-(c) are the input images taken by a Pentax K-7 camera with exposure bracketing mode of three exposures (-2EV, 0EV, 2EV). In Fig. 6 and 7, (d)-(f) are the results of the

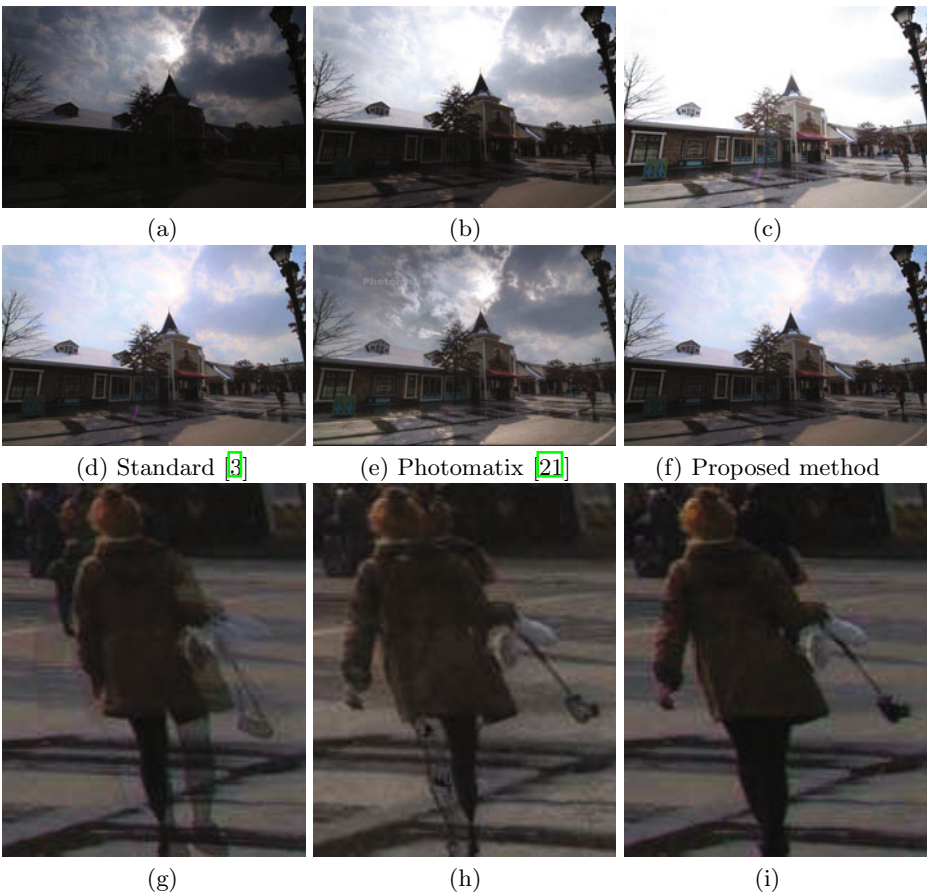


Fig. 7. Amusement park sequence. (a)-(c) are input LDR images, where (b) is the reference image. (d) Result of the standard method [3]. (e) Result of the Photomatix [21]. (f) Result of the proposed method. (g)-(i) are the magnified views of (d)-(f), respectively.

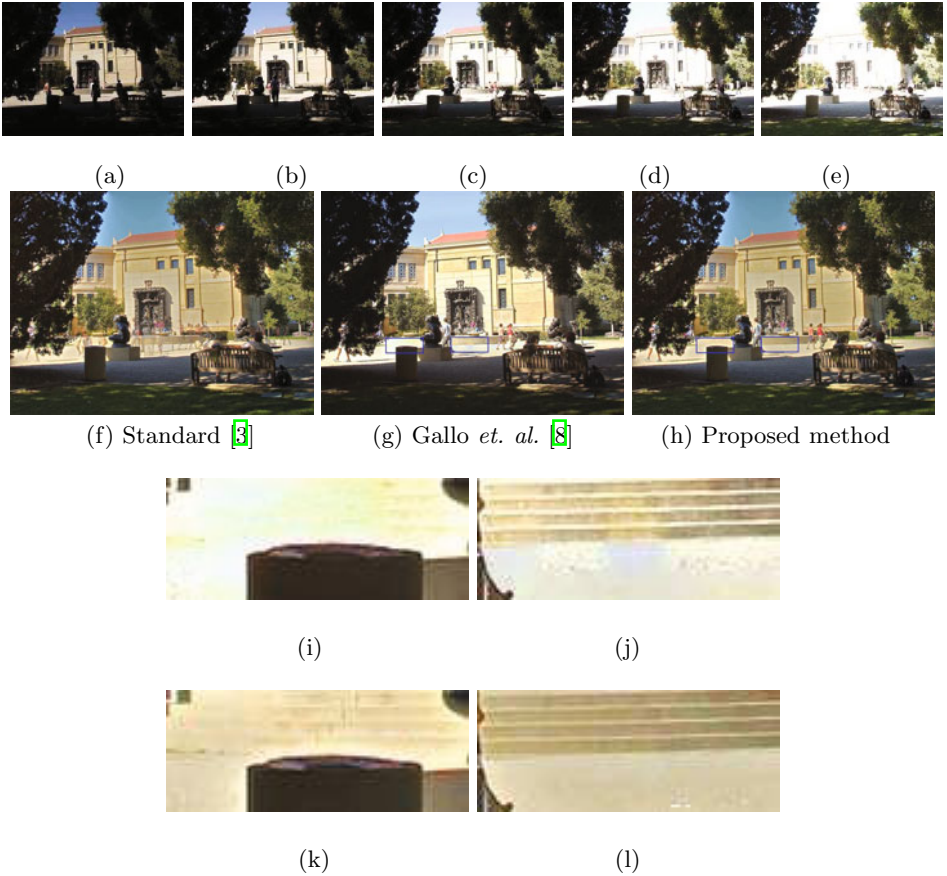


Fig. 8. Sculpture garden sequence. (a)-(e) are input LDR images, where (c) is the reference image. (f) Result of the standard method [3]. (g) Result of [8]. (h) Result of the proposed method. (i)-(j) are the magnified views of the blue rectangle regions in (g). (k)-(l) are the magnified views of the blue rectangle regions in (h).

standard method [3], the Photomatix [21], and the proposed method, respectively. For local movement regions, magnified views of (d)-(f) are shown in (g)-(i), respectively. As expected, we can observe that there are severe ghost artifacts in the standard method. Although the Photomatix reduces ghost artifacts a little bit, it can not completely eliminate them. Our method produces the most clean and ghost-free HDRIs even for severe local movement regions.

To further evaluate our method, we compared the results of our method with those of Gallo *et. al.* [8]. Fig. 8 and 9 show the comparison of ours with [8]. Two input sequences in Fig. 8 and 9 are from [8]. The standard method [3] severely suffers from ghost artifacts. Although [8] produces good results, it suffers from blending artifacts around block boundaries. On the contrary, our method produces more natural and clean HDRI with less color artifacts.

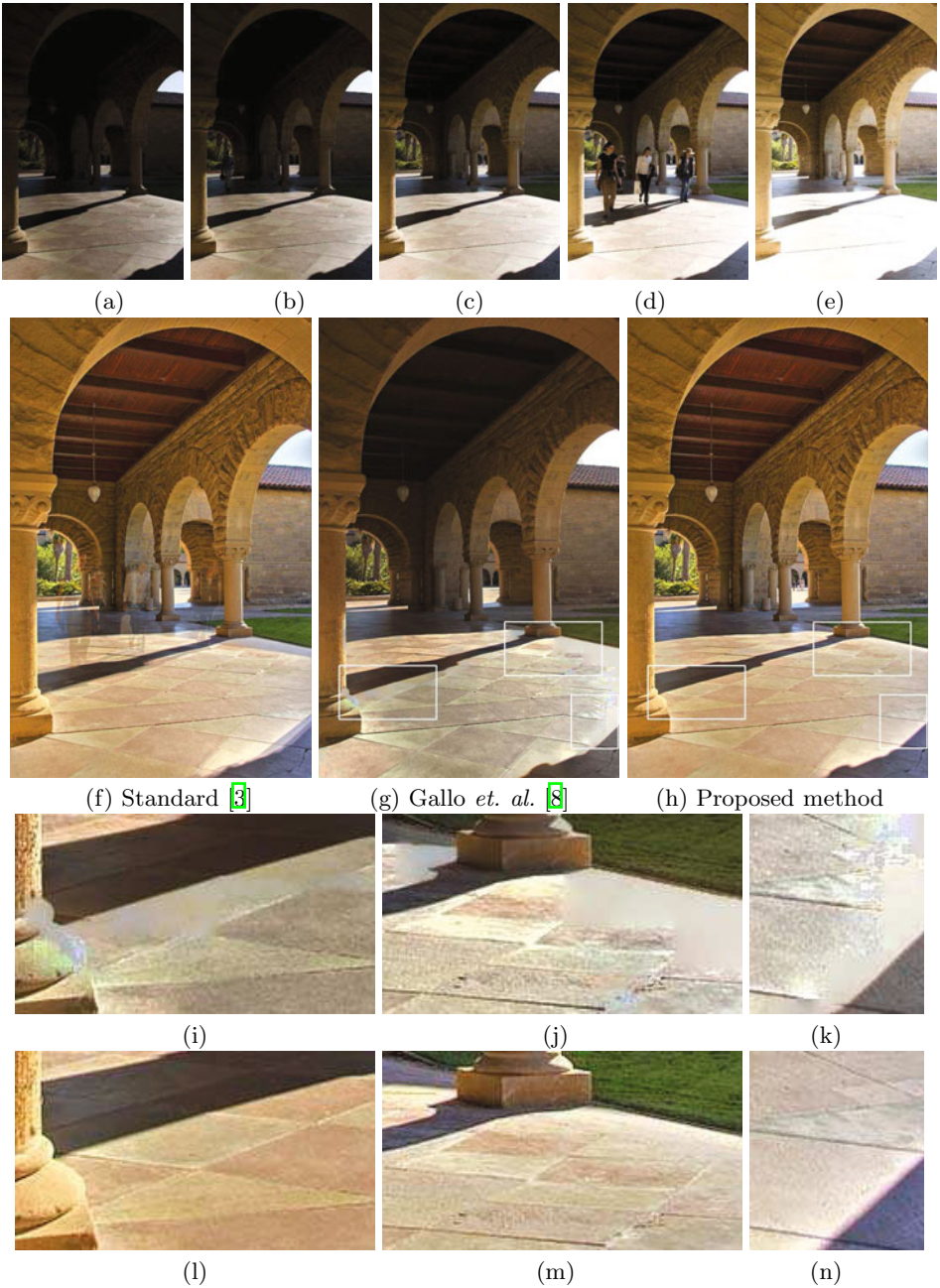


Fig. 9. Arch sequence. (a)-(e) are input LDR images, where (c) is the reference image. (f) Result of the standard method [3]. (g) Result of [8]. (h) Result of the proposed method. (i)-(k) are the magnified views of the white rectangle regions in (g). (l)-(n) are the magnified views of the white rectangle regions in (h).

5 Conclusion

In this paper, we have proposed an effective ghost elimination method for high dynamic range imaging using multiple exposure images of a dynamic scene. The proposed method is based on generalized weighted filtering using global intensity transfer functions between different exposures and refined radiance values. Our method does not need accurate ghost detection results which often include false positives or negatives, and also does not suffer from color artifacts such as visible seams between neighboring regions.

References

1. Reinhard, E., Ward, G., Pattanaik, S., Debevec, P.: High dynamic range imaging: Acquisition, display and image-based lighting. Morgan Kaufmann, San Francisco (2005)
2. Mann, S., Picard, R.W.: Being ‘undigital’ with digital cameras: Extending dynamic range by combining differently exposed pictures. Technical Report 323, M.I.T. Media Lab Perceptual Computing Section (1994)
3. Debevec, P.E., Malik, J.: Recovering high dynamic range radiance maps from photographs. In: Proc. of SIGGRAPH (1997)
4. Mitsunaga, T., Nayar, S.K.: Radiometric self calibration. In: Proc. of IEEE CVPR (1999)
5. Grossberg, M.D., Nayar, S.K.: What can be known about the radiometric response from images? In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2353, pp. 189–205. Springer, Heidelberg (2002)
6. Jacobs, K., Loscos, C., Ward, G.: Automatic high-dynamic range image generation for dynamic scenes. *IEEE Computer Graphics and Applications* 28, 84–93 (2008)
7. Grosch, T.: Fast and robust high dynamic range image generation with camera and object movement. In: Proc. of Vision, Modeling and Visualization, VMV (2006)
8. Gallo, O., Gelfandz, N., Chenz, W.C., Tico, M., Pulli, K.: Artifact-free high dynamic range imaging. In: Proc. of IEEE ICCP (2009)
9. Raman, S., Kumar, V., Chaudhuri, S.: Blind de-ghosting for automatic multi-exposure compositing. In: Proc. of SIGGRAPH Asia Sketches (2009)
10. Khan, E., Akyuz, A., Reinhard, E.: Ghost removal in high dynamic range images. In: Proc. of IEEE ICIP (2006)
11. Pedone, M., Heikkilä, J.: Constrain propagation for ghost removal in high dynamic range images. In: Proc. of VISAPP (2008)
12. Bogoni, L.: Extending dynamic range of monochrome and color images through fusion. In: Proc. of IEEE ICPR (2000)
13. Kang, S.B., Uyttendaele, M., Winder, S., Szeliski, R.: High dynamic range video. In: Proc. of SIGGRAPH (2003)
14. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. *IEEE Trans. PAMI* 23, 1222–1239 (2001)
15. Brown, M., Lowe, D.G.: Automatic panoramic image stitching using invariant features. *Int’l J. Computer Vision* 74, 59–73 (2007)
16. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM* 24, 381–395 (1981)

17. Kim, S.J., Pollefeys, M.: Robust radiometric calibration and vignetting correction. *IEEE Trans. PAMI* 30, 562–576 (2008)
18. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: *Proc. of IEEE ICCV* (1998)
19. Fattal, R., Lischinski, D., Werman, M.: Gradient domain high dynamic range compression. In: *Proc. of SIGGRAPH* (2002)
20. Reinhard, E., Stark, M., Shirley, P., Ferwerda, J.: Photographic tone reproduction for digital images. In: *Proc. of SIGGRAPH* (2002)
21. Photomatix, <http://www.hdrsoft.com/>

Pedestrian Recognition with a Learned Metric

Mert Dikmen, Emre Akbas, Thomas S. Huang, and Narendra Ahuja

Beckman Institute,
Department of Electrical and Computer Engineering,
University of Illinois at Urbana-Champaign
{mdikmen,eakbas2,t-huang1,n-ahuja}@illinois.edu

Abstract. This paper presents a new method for viewpoint invariant pedestrian recognition problem. We use a metric learning framework to obtain a robust metric for large margin nearest neighbor classification with rejection (i.e., classifier will return no matches if all neighbors are beyond a certain distance). The rejection condition necessitates the use of a uniform threshold for a maximum allowed distance for deeming a pair of images a match. In order to handle the rejection case, we propose a novel cost similar to the Large Margin Nearest Neighbor (LMNN) method and call our approach Large Margin Nearest Neighbor with Rejection (LMNN-R). Our method is able to achieve significant improvement over previously reported results on the standard Viewpoint Invariant Pedestrian Recognition (VIPeR [\[1\]](#)) dataset.

1 Introduction

Viewpoint invariant recognition of pedestrians is a problem that appears in numerous contexts in computer vision scenarios such as multi-camera tracking, person identification with an exemplar image or re-identification of an individual upon re-entering the scene after some time. This is a key problem and has been drawing attention in recent years with the advance of visual tracking and widespread deployment of surveillance cameras, which necessitated the need for continuous tracking and recognition across different cameras even with significant time and location differences. Our approach handles the long time delay case: recognition of the same individual without the temporal and spatial information associated with the images of the pedestrians. By learning an appropriate distance metric we achieve high recognition with high accuracy. Although we demonstrate it in the context of this problem, the learned metric is general and can be applied to aid data association in other tracking scenarios.

This paper assumes that the pedestrians in the scene has been successfully detected and consequently cropped. Pedestrian detection is an active research topic, but fortunately this problem is easier than the problem of general object detection and has been met with reasonable success with the emergence of several advanced methods in recent years. The relative success of pedestrian detection can be attributed to several limiting factors on the complexity of the

problem. Pedestrians are by definition upright people figures with limited configurations. Therefore template based approaches with a sliding window classifier produce favorable results [2,3]. In addition, there exists a number of strong and relatively easy to detect contextual cues, such as the presence of ground and other rigid objects (e.g., cars), which can be integrated into the decision process to significantly improve the detection performance [4].

Several attempts have been made for tackling the recognition problem in the context of matching pedestrians by their appearance only. Park et al. [5] perform recognition by matching color histograms extracted from three horizontal partitions of the person image. Hu et al. [6] have modeled the color appearance over the silhouette's principal axis. However, finding the principal axis requires robust background subtraction and is error prone in crowded situations. Matching spatio-temporal appearance of segments have been considered by Gheissari et al. [7]. Yu et al. [8] introduced a greedy optimization method for learning a distance function. Gray and Tao [1] defined the pedestrian recognition problem separate from multi-camera tracking context and provided a benchmark dataset (VIPeR, see Fig. 1) for standardized evaluation. Their method transforms the matching problem into a classification problem, in which a pair of images is assigned a positive label if they match (i.e., belong to the same individual) or negative label otherwise. This classifier is learned in a greedy fashion using Adaboost. The weak classifiers are decision stumps on individual dimensions of histograms of various features within a local rectangle in the person image. The rectangles span the entire horizontal dimension, while they are densely sampled vertically over all positions and sizes. Note that in the context of nearest neighbor classification, the $\{+1, -1\}$ labeling scheme of the matches vs non-matches



Fig. 1. Representative image pairs from the VIPeR dataset (images on each column are the same person). The dataset contains many of the challenges observed in realistic conditions, such as viewpoint and articulation changes as well as significant lighting variations.

creates a naturally unbalanced learning problem with N vs N^2 samples in two classes respectively ($N =$ number of training points). Also worth noting is that the two methods [8, 11], which learn the pairwise comparison function, achieve this through greedy optimization, which is not globally optimal and furthermore makes indirect use of covariances in the feature space. Our method is both globally optimal and also has an explicit covariance modeling of features.

The contributions of this paper are the following: (1) We apply a large margin nearest neighbor approach to the pedestrian recognition problem to achieve significantly improved results, (2) we define a novel cost function for learning a distance metric specifically for nearest neighbor problems with rejection. In addition we show that despite using only color as the appearance feature, our method is robust under significant illumination changes.

2 Metric Learning

In this section, we briefly introduce the metric learning framework of Weinberger and Saul [9] for large margin nearest neighbor (LMNN) classifier. The goal is to learn a Mahalanobis metric where the squared distances are denoted by:

$$\mathcal{D}_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{M} (\mathbf{x}_i - \mathbf{x}_j), \tag{1}$$

$\mathcal{D}_{\mathbf{M}}^{1/2}$ is a valid distance iff \mathbf{M} is a symmetric positive-semidefinite matrix. In this case \mathbf{M} can be factored into real-valued matrices as $\mathbf{M} = \mathbf{L}^\top \mathbf{L}$. Then, an equivalent form for [11] is

$$\mathcal{D}_{\mathbf{L}}(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{L}(\mathbf{x}_i - \mathbf{x}_j)\|^2. \tag{2}$$

LMNN learns a real-valued matrix \mathbf{L} that minimizes the distance between each training point and its K nearest similarly labeled neighbors (Eq. 3), while maximizing the distance between all differently labeled points, which are closer than the aforementioned neighbors' distances plus a constant margin (Eq. 4).

$$\varepsilon_{pull}(\mathbf{M}) = \sum_{i, j \rightsquigarrow i}^N \mathcal{D}_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j), \tag{3}$$

$$\varepsilon_{push}(\mathbf{M}) = \sum_{i, j \rightsquigarrow i} \sum_{k=1}^N (1 - y_{ik}) [1 + \mathcal{D}_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) - \mathcal{D}_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_k)]_+ \tag{4}$$

Here, y_{ik} is an indicator variable which is 1 if and only if \mathbf{x}_i and \mathbf{x}_j belong to the same class, and $y_{ik} = 0$ otherwise. The $j \rightsquigarrow i$ notation means that \mathbf{x}_j is one of the K similarly labeled nearest neighbors of \mathbf{x}_i (i.e., \mathbf{x}_j is a target neighbor of \mathbf{x}_i). Note that for ε_{pull} to be a continuous and convex function, it is necessary that the K target neighbors of each training sample be fixed at the initialization. In practice they are determined by choosing the K nearest neighbors by Euclidean distance.

The \mathbf{x}_k in Eq. 4 for which $y_{ik} = 0$ are called the impostors for \mathbf{x}_i . The expression $[z]_+ = \max(z, 0)$ denotes the standard hinge loss. Although this hinge loss is not differentiable at $z = 0$, we did not observe any convergence issues. Nevertheless it is always possible to replace the standard hinge loss with a smooth approximation [10].

The affine combination of ε_{pull} and ε_{push} through the tuning parameter μ (Eq. 5) defines the overall cost, which essentially maximizes the margin for K nearest neighbor classifier by pulling together same-labeled points and repelling differently-labeled ones (impostors).

$$\begin{aligned} \varepsilon_{\text{LMNN}}(\mathbf{M}) = & (1 - \mu) \sum_{i, j \rightsquigarrow i} \mathcal{D}_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \\ & + \mu \sum_{i, j \rightsquigarrow i} \sum_{k=1}^N (1 - y_{ik}) [1 + \mathcal{D}_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) - \mathcal{D}_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_k)]_+. \end{aligned} \quad (5)$$

2.1 Nearest Neighbor with Rejection

In this section we introduce our LMNN-R framework for doing K nearest neighbor classification with the option of rejection. As a practical example for this problem, consider the person re-identification task, where given an image of a pedestrian, one would like to determine whether the same person is in the current scene or not. The target set of the people in the scene may not contain the query person. One way to adapt the nearest neighbor classifier to the problem of re-identification is to adopt a universal threshold (τ) for maximum allowed distance for matching image pairs. If the distance of the nearest neighbor of the query in the target set is greater than τ , one would deem that the query has no match in the target set (rejection). Conversely, if there is a nearest neighbor closer than τ , then it is called a match. What we have just described is the 1 nearest neighbor with rejection problem. This problem can be extended to K nearest neighbor case, in which a label is assigned through majority voting of P nearest neighbors within τ , where $P \leq K$. If $P = 0$ the classifier will refuse to assign a label.

The introduction of the option to refuse label assignment necessitates a distance metric that allows the use of a global threshold in all localities of the feature space. One method would be to assume unimodal class distributions as proposed by Xing et al. [11]. Their objective function maximizes the distance between all sample pairings with different labels, while a constraint is imposed on the pairs of similarly labeled points to keep them closer than a universal distance. This model was proposed for learning a distance metric for k-means clustering. It does not directly apply to our problem formulation. One drawback is the situation when similarly labeled samples do not adhere to a unimodal distribution (e.g., two islands of samples with same labels). Another problem is the lack of margin in their formulation, which is essential for good generalization

¹ All reported experiments in this paper use $\mu = 0.5$ for both LMNN and LMNN-R.

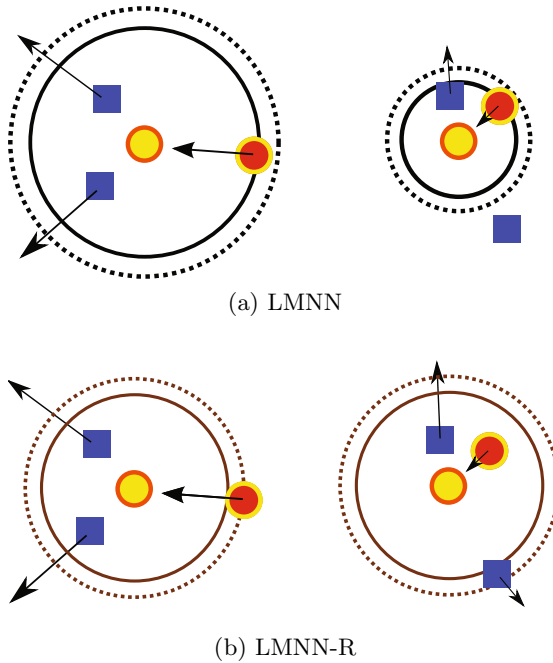


Fig. 2. Illustration contrasting our proposed approach with [9]. Note that the point configurations for a) and b) are the same. For a given training point (yellow), the target neighbor (red) is pulled closer, while the impostors (blue) are pushed away. a) To determine the impostors, the LMNN cost function uses a variable distance from the training point depending on the proximity of the target neighbors; b) LMNN-R on the other hand, forces the impostors out of a universal distance from the training point, while simultaneously attracting target neighbors.

performance in classification. A cost function, which emphasizes local structure is more suitable in our case.

We adopt the LMNN cost function (Eq. 5), which minimizes the distance between each training point and its K nearest similarly labeled neighbors (Eq. 3), while maximizing the distance between all differently labeled points, which are closer than the aforementioned neighbors' distances plus a constant margin (Eq. 4). The margin imposes a buffer zone to ensure good generalization. It is this local property that makes the LMNN metric learning very suitable to nearest neighbor classification. Note that the distance to determine the impostors is varying for each training point \mathbf{x}_i (Eq. 4). We replace this with a universal distance: the average distance of all K nearest neighbor pairs in the training set (Eq. 6). LMNN-R cost function forces the closest impostors of a training point to be at least a certain distance away, determined by this average which is only weakly affected by where its own K nearest neighbors are (Fig. 2). The net effect of this modification is that now we can use a universal threshold on pairwise distances for determining rejection, while still approximately preserving the local

structure of the large margin metric learning. The only requirement for the loss function to be convex is that the K nearest neighbor structure of the training points need to be pre-defined. However, extensions such as multi-pass optimization [9] proposed to alleviate this problem for LMNN apply to LMNN-R also.

$$R = \frac{1}{NK} \sum_{m,l \rightsquigarrow m} \mathcal{D}_{\mathbf{M}}(\mathbf{x}_m, \mathbf{x}_l) \tag{6}$$

$$\varepsilon_{\text{LMNN-R}}(\mathbf{M}) = (1 - \mu)\varepsilon_{\text{pull}}(\mathbf{M}) + \mu\varepsilon_{\text{push}}^*(\mathbf{M}) \tag{7}$$

$$\varepsilon_{\text{push}}^*(\mathbf{M}) = \sum_{i=1}^N \sum_{k=1}^N (1 - y_{ik}) \left[1 + \frac{1}{NK} \left(\sum_{m,l \rightsquigarrow m} \mathcal{D}_{\mathbf{M}}(\mathbf{x}_m, \mathbf{x}_l) \right) - \mathcal{D}_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_k) \right]_+ \tag{8}$$

The LMNN-R cost (Eq. 7) can be minimized as a semidefinite program, which is formulated by writing $\varepsilon_{\text{push}}^*$ as a constraint through the introduction of slack variables, or it can be minimized by following the gradient directly and projecting \mathbf{M} back to the semidefinite cone at each iteration (iterative sub-gradient projection as in [9]).

3 Experiments

We demonstrate the performance of our method on the VIPeR dataset [12] which is a specifically constructed dataset for the viewpoint invariant pedestrian recognition problem. This dataset contains images of 632 unique pedestrians and a total of 1264 images composed of two views per pedestrian seen from different viewpoints. The images are captured outdoors under uncontrolled lighting. Therefore there is a great deal of illumination variance in the dataset, including between the images belonging to the same pedestrian (e.g., the first and last columns in Figure 1). Compared to the previously available datasets (see [1]), the VIPeR dataset has many more unique subjects and contains a higher degree of viewpoint and illumination variation, which makes it realistic and more challenging (Figure 1).

3.1 Methodology

As done in [1], we randomly split the set of pedestrians into two halves: training and testing. The LMNN and LMNN-R frameworks learn their respective distance metric using the training set. For testing, each image pair of each pedestrian in the test set is randomly split to query and target sets. The results are generated using the pairwise distance matrix between these query and target subsets of the images in the test set. For thoroughness, we report our results as an average over 10 train-test splits. When reporting an average is not appropriate, we report our best result out of the 10 splits.

We follow the same evaluation methodology of [1] in order to compare our results to theirs and other benchmark methods. We report results in the form of cumulative matching characteristics curve (CMC), re-identification rate curve and expected search time by a human operator. In addition, we also provide an average receiver operator characteristic curve to demonstrate the improvement of the LMNN-R method over LMNN for automated recognition.

3.2 Image Representation

The images in the dataset are 128 pixels tall and 48 pixels wide. We use color histograms extracted from 8×24 rectangular regions to represent the images. The rectangular regions are densely collected from a regular grid with 4 pixel spacing in vertical and 12 pixel spacing in horizontal direction. This step size is equal to half the width and length of the rectangles, providing an overlapping representation.

For the color histograms, we use RGB and HSV color spaces and extract 8-bin histograms of each channel separately. We tried several combinations for all of the mentioned parameters found that these numbers worked reasonably well through our preliminary experiments. We concatenate the histograms extracted from an image and obtain a feature vector of size 2232 for RGB and HSV representations each. The combined representation is simply the concatenation of these two. Dimension reduction through PCA is applied to these high-dimensional vectors to obtain subspaces of specific dimensionality. This step is necessary to reduce redundancy in the color based representation and to filter out some of the noise. The reported results are obtained with 20, 40 and 60 dimensional representations. We have observed that we get diminished returns above 60 dimensions.

To account for the illumination changes we experiment with a simple color correction technique where each RGB channel of the image is histogram-equalized independently to match a uniform distribution as close as possible in ℓ_1 norm. Since in the cropped images, a significant number of the pixels belong to the pedestrian, this is a reasonable way of performing color correction. We also experimented with brightness and contrast correction methods, as well as histogram equalizing the V channel of the HSV images. However, they were not able to perform as good as the described RGB histogram equalization method.

3.3 Results

Recognition. We present the recognition performances as CMC curves in Figure 3. This curve, at rank score k , gives us the percentage of the test queries whose target (i.e. correct match) is within the top k closest match. As it is not appropriate to take the average of CMC curves over different random splits of the dataset, we report the CMC of a single split where the normalized area under the curve is maximum. This corresponds to using “RGB+HSV” features reduced to 60 dimensions via PCA and using our proposed approach LMNN-R. We outperform

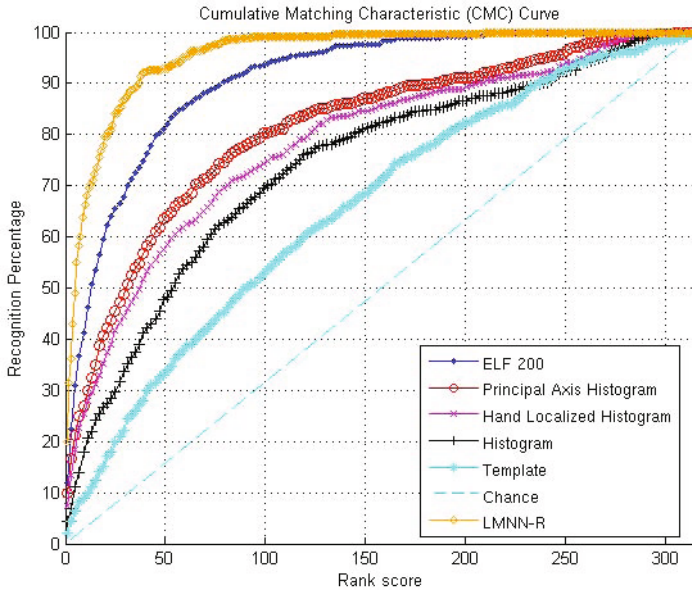


Fig. 3. Cumulative matching characteristics (CMC) curve for our method and others'. This result is obtained using a combined HSV and RGB representation in a 60 dimensional subspace learned with PCA.

all previously reported results². An explanation of the methods used to obtain these previous results is as follows. “Chance” refers to random matching, “Template” refers to pixelwise sum-of-squared distances matching. “Histogram” and “Hand Localized Histogram” refer to the method by Park et al. [5], and “Principal Axis Histogram” refers to the method of Hu et al. [6]. “ELF 200” (or just “ELF” in the remaining of the text) refers to the work of Gray et al. [1].

CMC curves can be summarized using the “expected search time” measure defined in [1]. Assuming a human operator reviews a query image’s closest matches sequentially according to their distance from the query. Assuming an average review time of 1s per image, the total expected search time for finding the correct match would be the average rank of the target. Our method’s expected target rank is 23.7 which is an improvement of over 15% with respect to the state-of-the-art 28.9 (see Table II).

To evaluate the performance of LMNN and LMNN-R over all different combinations of parameter and feature choices, we use the normalized area under the CMC curves. Table 2 shows the mean and standard deviation of these values over 10 random splits of the dataset. Best results are obtained using RGB and HSV together on original (non-color corrected) images. RGB alone performs the worse than HSV alone, which is expected because HSV is more robust to variations in intensity of the lighting.

² Results of other methods are from [1] as a courtesy of D. Gray.

Table 1. Expected search times for LMNN-R and other methods

Method	Expected Search Time (in seconds)
Chance	158.0
Template	109.0
Histogram	82.9
Hand Localized Histogram	69.2
Principal Axis Histogram	59.8
ELF	28.9
LMNN-R	23.7

Since the dataset has a significant degree of illumination variation, one expects that color correction should help increase the matching accuracy. While this is true for the plain ℓ_2 norm (i.e. no learning), it is not the case for learned metrics of LMNN and LMNN-R. A possible explanation for this can be made by realizing that the histogram equalization process is a non linear transformation of the data. While improving the performance of the marginal cases for simple matching by Euclidean distances, this procedure may affect the average transformation that image pairs undergo in realistic scenarios, such that this transformation cannot be reliably modeled by LMNN and LMNN-R methods anymore. Therefore we suggest letting the learning algorithm handle the color correction issues.

For the number of reduced dimensions, 60 is slightly better than 40. And LMMN-R gives slightly better results than LMNN in general.

In the previous re-identification experiments, we assume that the target set will have a match for the query image. This is not the case in many practical scenarios as often it is not known whether the query person is in view. Therefore

Table 2. Table of results averaged over 10 random splits of the dataset. 20, 40 and 60 denote the number of dimensions (of the reduced subspace found by PCA) used, L_2 refers to the regular ℓ_2 norm which, in our case, corresponds to “no learning”. “corr’d” means “color corrected” and “orig” indicates that no modification was done to the original image. We obtain our best average results using RGB and HSV together on original images with the proposed learning approach LMNN-R. The overall best result, i.e. the one given in Figure 3, has a normalized area of 95.88 under its CMC curve, which is comparable to the average results.

		RGB+HSV		HSV		RGB	
		corr'd	orig	corr'd	orig	corr'd	orig
20	L_2	76.61 ± 0.88	72.54 ± 0.77	80.09 ± 0.59	77.97 ± 0.81	67.85 ± 1.13	60.63 ± 0.79
	LMNN	91.81 ± 0.39	93.46 ± 0.36	92.11 ± 0.47	92.90 ± 0.34	82.06 ± 0.69	86.39 ± 0.72
	LMNN-R	92.14 ± 0.37	93.59 ± 0.37	92.35 ± 0.47	92.87 ± 0.51	82.47 ± 0.83	86.63 ± 0.68
40	L_2	77.48 ± 0.87	73.73 ± 0.81	80.79 ± 0.66	78.89 ± 0.80	68.73 ± 1.11	60.90 ± 0.92
	LMNN	92.68 ± 0.44	94.54 ± 0.42	92.82 ± 0.33	94.40 ± 0.32	83.81 ± 1.27	87.14 ± 0.86
	LMNN-R	93.13 ± 0.48	94.76 ± 0.47	93.04 ± 0.45	94.64 ± 0.43	84.71 ± 1.24	87.49 ± 0.92
60	L_2	77.85 ± 0.86	74.14 ± 0.79	80.97 ± 0.67	79.17 ± 0.80	68.83 ± 0.91	61.20 ± 0.91
	LMNN	92.27 ± 0.50	94.67 ± 0.55	92.52 ± 0.28	94.54 ± 0.29	84.23 ± 0.63	87.56 ± 1.01
	LMNN-R	92.56 ± 0.53	94.95 ± 0.46	92.62 ± 0.43	94.69 ± 0.37	84.94 ± 0.57	87.79 ± 1.04

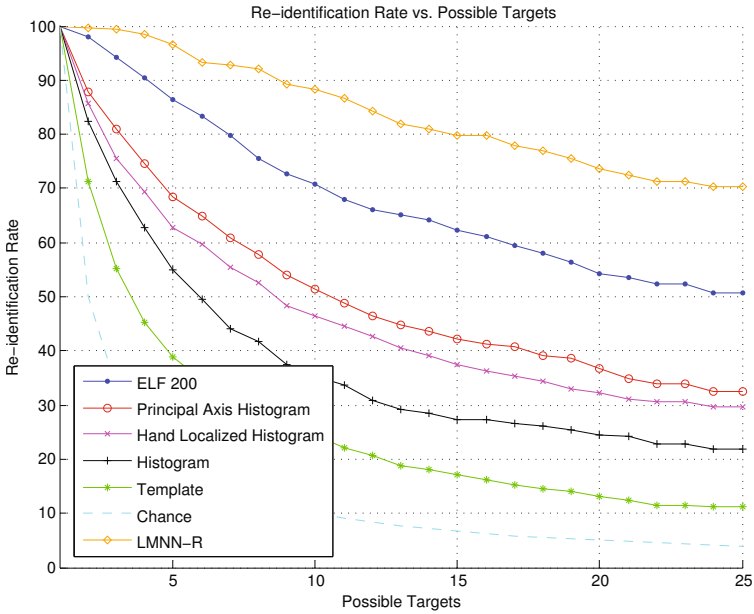


Fig. 4. Re-identification rate vs. the number of targets for our method and others

we also show the receiver operator characteristic curve (ROC) for such kind of cases where one would like to detect the query pedestrian in a target set of pedestrians. The detection performance is measured by comparing the true positive rate vs. the false positive rate, which shows for a given recall rate (true positive), what fraction of non matching images in the target set will be returned as false positives. Due to the universal threshold, the LMNN-R method was able to outperform LMNN by about 1% at a false positive rate of 10% (Fig. 5).

Re-identification. This is another measure for evaluating the performance of pedestrian matching methods. It is the probability of finding a correct match as a function of the number of possible targets. A formal definition could be found in [12]. Figure 4 shows the re-identification rates of our method and the previous methods.

Execution times. We implemented LMNN and LMNN-R in MATLAB³ and although we have not employed the active set method which was designed to make LMNN more efficient (described in [9]), our code runs reasonably fast in practice. For the VIPeR dataset, a typical training session takes 160 seconds and finding the target of a query pedestrian takes only 1.2 milliseconds on a 2GHz Intel Core2-Duo PC.

³ The MATLAB code for LMNN and LMMN-R optimization as well as replicating the experiments in the paper is available in the supplementary material of the paper.

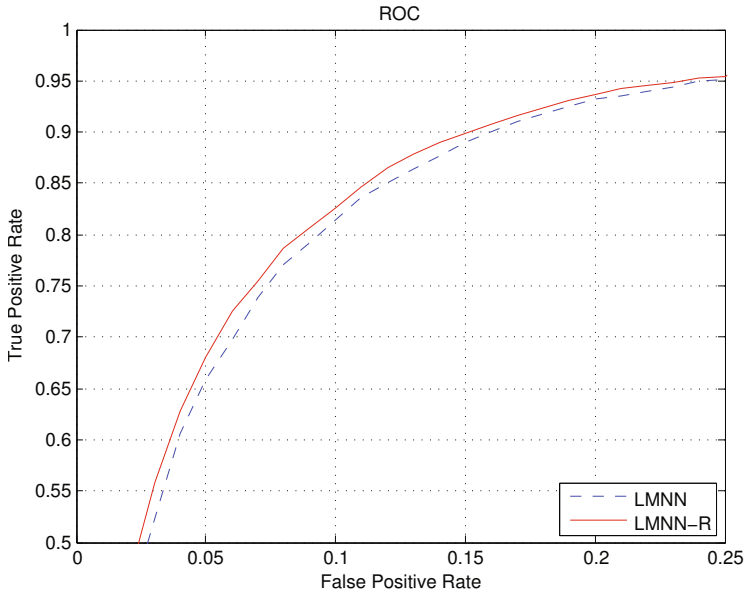


Fig. 5. The receiver operator characteristic curve showing the true positive vs the false positive rate of our system

4 Conclusions

We have applied a large margin nearest neighbor (LMNN) approach to viewpoint invariant pedestrian recognition problem. Also, we proposed a new variant of LMNN called large margin nearest neighbors classification with rejection (LMNN-R) to obtain a classifier with the option of rejecting unfamiliar matches. Using only color histograms as features, these methods achieved significant improvement over previously reported results on a benchmark dataset. Experimental results suggest that our LMNN-R formulation to metric learning is able to achieve improved results over LMNN. Color correction improved the matching accuracy when Euclidean distance is used to compare images (i.e. no learning). However, this was not the case for LMNN and LMNN-R which suggests that these supervised learning approaches are more robust in handling illumination changes than color correction alone.

References

1. Gray, D., Tao, H.: Viewpoint invariant pedestrian recognition with an ensemble of localized features. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 262–275. Springer, Heidelberg (2008)
2. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, p. 886 (2005)

3. Tuzel, O., Porikli, F., Meer, P.: Pedestrian detection via classification on riemannian manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30, 1713–1727 (2008)
4. Hoiem, D., Efros, A., Hebert, M.: Putting objects in perspective. *International Journal of Computer Vision* 80, 3–15 (2008)
5. Park, U., Jain, A., Kitahara, I., Kogure, K., Hagita, N.: Vise: visual search engine using multiple networked cameras. In: *Proceedings of the 18th International Conference on Pattern Recognition (ICPR 2006)*, vol. 3, pp. 1204–1207 (2006)
6. Hu, W., Hu, M., Zhou, X., Tan, T., Lou, J., Maybank, S.: Principal axis-based correspondence between multiple cameras for people tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28, 663–671 (2006)
7. Gheissari, N., Sebastian, T., Hartley, R.: Person reidentification using spatiotemporal appearance. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 1528–1535 (2006)
8. Yu, J., Amores, J., Sebe, N., Radeva, P., Tian, Q.: Distance learning for similarity estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30, 451–462 (2008)
9. Weinberger, K., Saul, L.: Distance metric learning for large margin nearest neighbor classification. *The Journal of Machine Learning Research* 10, 207–244 (2009)
10. Rennie, J.D.M., Srebro, N.: Fast maximum margin matrix factorization for collaborative prediction. In: *Proceedings of the 22nd International Conference on Machine Learning, ICML 2005*, pp. 713–719 (2005)
11. Xing, E., Ng, A., Jordan, M., Russell, S.: Distance metric learning with application to clustering with side-information. In: *Advances in Neural Information Processing Systems*, pp. 521–528 (2003)
12. Gray, D., Brennan, S., Tao, H.: Evaluating appearance models for recognition, reacquisition, and tracking. In: *Proc. IEEE Int'l Workshop on Performance Evaluation for Tracking and Surveillance, PETS* (2007)

A Color to Grayscale Conversion Considering Local and Global Contrast

Jung Gap Kuk, Jae Hyun Ahn, and Nam Ik Cho

Institute of New Media & Communications (INMC)
Dept. of Electrical Engineering, Seoul National University
Seoul, 151-744, Korea
{jg-kuk, jhahn}@ispl.snu.ac.kr, nicho@snu.ac.kr

Abstract. For the conversion of a color image to a perceptually plausible grayscale one, the global and local contrast are simultaneously considered in this paper. The contrast is measured in terms of gradient field, and the energy function is designed to have less value when the gradient field of the grayscale image is closer to that of original color image (called target gradient field). For encoding both of local and global contrast into the energy function, the target gradient field is constructed from two kinds of edges : one that connects each pixel to neighboring pixels and the other that connects each pixel to predetermined landmark pixels. Although we can have exact solution to the energy minimization in the least squares sense, we also present a fast implementation for the conversion of large image, by approximating the energy function. The problem is then reduced to reconstructing a grayscale image from the modified gradient field over the standard 4-neighborhood system, and this can be easily solved by the fast 2D Poisson solver. In the experiments, the proposed method is tested on various images and shown to give perceptually more plausible results than the existing methods.

1 Introduction

Most of photos nowadays are taken in color and many of printed matters are also produced in colors. Of course, the colorful material has many advantages over the black-and-white (BW) ones. However, color printing requires more cost, time, and energy, and thus it is desirable to print in BW mode or with BW printers. For printing a colorful material in the BW printers, the conversion from N_c^3 colors to N_g grayscale values should be performed where $N_c = N_g = 256$ in most cases. This process entails the loss of information, specifically the chromatic contrast can be lost in the converted grayscale image. For example, the edges perceived in a color image cannot be seen in the converted grayscale image.

For perceptually plausible conversion, i.e., for keeping the chromatic difference as much as possible in the converted grayscale image, many algorithms have been introduced. The most straightforward method is to find a fixed linear or non-linear mapping from N_c^3 colors to N_g grayscale values [6–8]. However, since too

many colors are mapped to a single grayscale value, these methods often fail to convey the chromatic contrast to the grayscale image. The other approaches are to consider the spatial relationship to keep the chromatic difference [4, 12]. This can be considered as a locally varying mapping, enabled by formulating a cost function to be minimized. By the locally varying mapping, the pixels with the same color, but located apart from each other, can be given different grayscale values and thus there is less chance of losing the chromatic information than the fixed mapping method. However, these methods suffer from unwanted contours because they place too much emphasis on reproducing perceived chromatic difference. That is, almost the same color can be given too much grayscale difference, and thus the object with smoothly changing color can experience the contour effect.

In this paper, we propose a new color to grayscale conversion algorithm which well reproduces the chromatic contrast while avoiding unwanted contours. For this, we note that the global contrast (contrast between the pixels apart from each other) as well as the local contrast is important. For encoding these global and local chromatic distance simultaneously, we prepare a metric that measures the signed distance between the colors, and construct a target gradient field over a graph where two kinds of edges are defined : one connects a pixel to neighboring pixels and the other that connects a pixel to predetermined landmark pixels with dominant colors found by color quantization [5]. The energy is then designed so that gradient field of grayscale image is as close as to target gradient field when it achieves minimum. In the least squares sense, the energy minimization is reduced to a sparse linear system, which can be solved directly [3] or iteratively [14]. Although these solvers are quite fast and alleviate the memory requirement problem, it is still problematic when dealing with over 1 mega pixels. Hence, we also present a fast implementation for the large images, by approximating the energy. The approximated energy does not have any dependencies on landmark pixels and the problem is reduced to the reconstruction of the grayscale image from the gradient field over standard 4-neighborhood system. This is efficiently solved by fast 2D Poisson solver and we can deal with over 1 mega pixels in a few seconds.

The rest of paper is organized as follows. In section 2 we briefly review the previous algorithms, and the details of the proposed method is explained in section 3. In subsection 3.1, we formulate the energy, and then present a fast approximate solution in subsection 3.2. In section 4 we verify the proposed method by testing on various images, and finally we conclude this paper in section 5.

2 Review of Previous Methods

The existing methods can be categorized into two approaches : finding a fixed mapping from the color space to the grayscale and an energy function minimization for finding a locally varying mapping.

2.1 Fixed Mapping

First of all, a linear fixed mapping was proposed in [8]. This method converts RGB to YPQ and then linearly combines luminance component Y and chromatic information. The chromatic information is obtained by projecting chrominance components P and Q to the predominant chromatic axis. This scheme adds more chromatic information to the color which was to lose the contrast otherwise. Note that this method estimates contrast loss between two pixels, where a location of the pixel to the other is randomly selected so that the global contrast is implicitly considered. There has also been a method that finds the parameterized fixed mapping function for the non-linear mapping [7]. The parameters are estimated by minimizing the cost function which is designed to reproduce chromatic differences between the neighboring pixels. The method is also extended to temporally coherent conversion of streaming videos. Also, there is an image-independent global mapping method [6]. This method explores several chromatic lightness metrics which consider the Helmholtz-Kohlrausch effect and applies variable-achromatic-colour approach in [9] among them. Since this fixed mapping is image-independent, the result is locally enhanced to restore lost discontinuities.

Although the methods in this approach show good performance, the chromatic differences in color image are sometimes lost, since many colors are mapped to a single value.

2.2 Variable Mapping by Energy Minimization

For finding the area dependent mapping, the method in [4] measures signed distances between all the pixel pairs in color image and formulates an energy function to reproduce them in grayscale image. Since the energy in this method is defined for all the pixel pairs, the original version of this method needs very high complexity, $O(N^4)$. There is also a fast version of this method as a Photoshop plugin (<http://www.e56.de/c2g.php>), but it sometimes causes artifacts. Instead of considering contrasts between all the pixel pairs, the method in [12] is focused on preserving contrasts among the selected 256 landmark colors. Specifically, this method first makes a mapping from 256 landmark colors to the 256 grayscale values by minimizing the energy in a similar form to [4] and the remnant colors in the image are then interpolated based on the given mapping.

These methods are technically well accepted, but unwanted contours are sometimes produced because they place much emphasis on reproducing global contrast. Our algorithm also lies in this category (variable mapping) and we design an algorithm to avoid these unwanted contours by encoding both of local and global contrast into the energy function in a controlled manner.

3 Details of the Proposed Method

In this section, we present the proposed method in detail. We first formulate the energy function for the variable mapping in subsection 3.1. Then, we present



Fig. 1. The result of popularity method [5] when $M=30$ and $b=3$. (a) Original image. (b) Reconstructed image from 30 colors.

the fast implementation for the conversion of large images, by approximating the energy function in subsection 3.2. For the intuitive understanding, we also give a graphical interpretation of the energy approximation in subsection 3.2.

3.1 Energy Formulation

In order to design an energy function that encodes both of local and global contrast, we begin with defining a metric that measures a scalar distance between two colors and finding the landmark pixels that are used as references for finding global contrast.

There have been numerous color metrics based on coloroid system [10], Lab space [4, 7, 10, 12] and YPQ space [8]. Among them, we choose a simple metric explored in [10], which is defined over the Lab space as

$$\delta_{ij} = \sqrt[3]{(L_i - L_j)^3 + w_a(a_i - a_j)^3 + w_b(b_i - b_j)^3} \tag{1}$$

where δ_{ij} denotes the distance between two colors (L_i, a_i, b_i) and (L_j, a_j, b_j) , and w_a and w_b are free parameters for adjusting the contributions of a and b channels to the distance respectively. Selecting landmark pixels in a given color image is based on the popularity method [5], which is one of the color quantization methods. This method clips each color sample to $3N_b$ bits (N_b bits for each channel), generates a histogram with 2^{3N_b+1} bins and chooses M the most populated bins. Averaging the colors in each of the chosen bins and searching the pixel position with the closest color to the mean, we have M dominant colors and their pixel positions Q . Throughout this paper, we set $b=3$ and $M=30$. As shown in an example of Fig. 1(b), selected colors by the popularity method with this parameter setting represent the original color image very well.

With the above metric and landmark colors, the proposed energy is formulated as

$$f(\mathbf{g}) = (1 - \lambda) \sum_{(i,j) \in E} (g_i - g_j - \delta_{ij})^2 + \lambda \sum_{i \in V} \sum_{k \in Q} (g_i - g_k - \delta_{ik})^2 \tag{2}$$

where \mathbf{g} is a vector representation of grayscale image (g_i is the i th element of \mathbf{g}), E is a set of edges over standard 4-neighborhood system, V is a set of whole pixels and λ is a control parameter. In (2), the first term is responsible for preserving the local contrast between neighboring pixels, and the second for the preservation of global contrast between a pixel and landmark pixels.

Note that one of the previous methods [4] finds the grayscale image by considering the distance between all the pixel pairs. This method is to minimize the function which can be written in the form of (2) as

$$\tilde{f}(\mathbf{g}) = \sum_{i \in V} \sum_{j \in Q} (g_i - g_j - \delta_{ij})^2 \tag{3}$$

with $Q = V$. However, considering all the available pixel pairs to preserve global contrast as in (3) sometimes produces unwanted contours, whereas our method can avoid the contours owing to the small size of Q . Also, since our energy function is separated into local term (first term of (2)) and global term (second term), we can balance the local contrast with the global contrast, whereas such an energy formulation in (3) cannot. The balance is adjusted by the control parameter λ and the effect of λ will be discussed in the experiment.

To find the solution that minimizes the energy function $f(\mathbf{g})$, we differentiate it with respect to g_i and set it zero as

$$(1 - \lambda) \sum_{j \in N_i} (g_i - g_j - \delta_{ij}) + \lambda \sum_{k \in Q} (g_i - g_k - \delta_{ik}) = 0, \quad i \in V \tag{4}$$

where N_i is a set of neighbors of the i th pixel. Aggregating all the linear equations to make them in a vector form, we finally have the following linear system :

$$((1 - \lambda)\mathbf{S} + \lambda\mathbf{M}\mathbf{I} + \lambda\mathbf{P})\mathbf{g} = \mathbf{b}, \tag{5}$$

where \mathbf{S} represents a symmetric matrix responsible for the first term in (4), \mathbf{I} is an identity matrix, \mathbf{b} is a vector whose element b_i is $(1 - \lambda) \sum_{j \in N_i} \delta_{ij} + \lambda \sum_{k \in Q} \delta_{ik}$ and \mathbf{P} is the matrix which satisfies that $\mathbf{P}(:, i) = -1$, if $i \in Q$ and $\mathbf{P}(:, i) = 0$, otherwise. This linear system is sparse and it can be solved directly [3] or iteratively [14]. Note that the solution of (5) is not unique because shifting \mathbf{g} by any constant c has the same energy. Hence, we find the constant such that the solution is close to the lightness image L .

As expected, the proposed method well preserves contrasts as shown in Fig. 2 (c), while previous fixed mapping based method fails to as shown in Fig. 2 (b). In this result, the proposed method is compared with a decolorize method in [8] as it earned the top overall score in the review of color to gray conversion by Čadík [2]. Also the proposed method does not produce unwanted contours as shown in Fig. 3 (h) whereas the previous energy minimization based methods produce the contours as shown in Fig. 3 (f) and Fig. 3 (g).

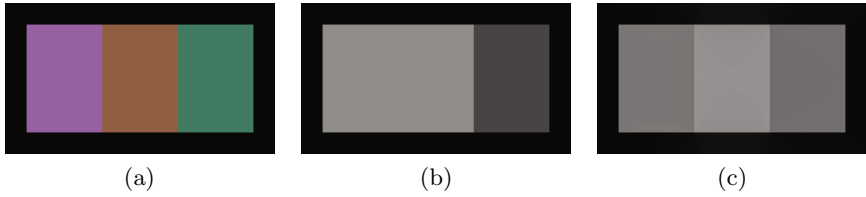


Fig. 2. (a) Color image. (b) The result of [8]. Edge between first and second column is missing because two colors in contact are mapped to the same gray value. (c) The result of the proposed method. Perceived chromatic contrast is well reproduced in grayscale image.

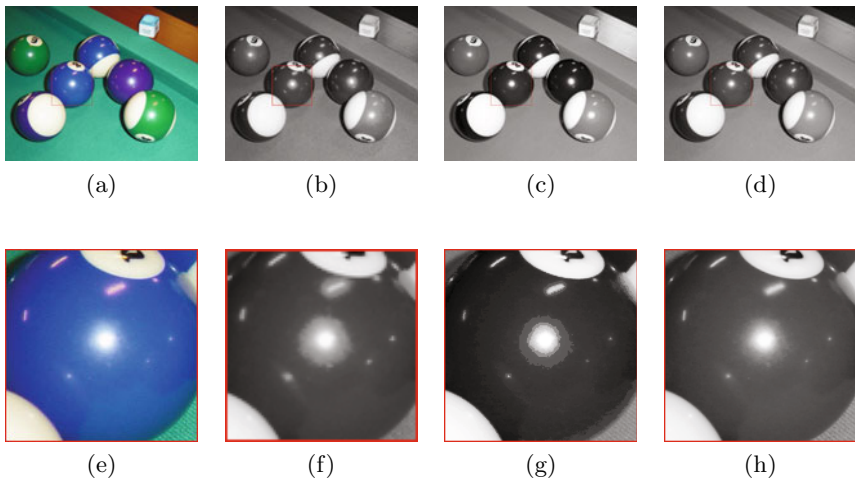


Fig. 3. (a) Color image. (b) The result of [4]. (c) The result of [12]. (d) The result of proposed method. (e-h) The enlarged images of selected area in (a-d). The methods in [4] and [12] produce unwanted contours.

3.2 Fast Approximate Solution

Solving the large sparse linear system requires much computations when dealing with over 1 mega pixels. Hence in this subsection, we present a fast implementation by approximating the energy function.

The main computational cost in the proposed algorithm is due to the fact that the linear system of (5) is not symmetric, i.e., the asymmetry of the matrix \mathbf{P} . \mathbf{P} is related to the second term of the energy in (2) and we modify this term to make the linear system symmetric. Applying several algebraic steps to the second term of the energy in (2) gives

$$\sum_{i \in V} \sum_{k \in Q} (g_i - g_k - \delta_{ik})^2 = \sum_{i \in V} \sum_{k \in Q} (g_i - g_n + g_n - g_k - \delta_{ik})^2 \tag{6}$$

$$= \sum_{i \in V} \sum_{k \in Q} \frac{1}{|N_i|} \sum_{j \in N_i} (g_i - g_j + g_j - g_k - \delta_{ik})^2 \tag{7}$$

where an auxiliary variable g_n is introduced without loss of generality. g_n can be arbitrarily chosen as a neighbor of the i th pixel as in (7). In (7), we make an assumption that the grayscale difference between a pixel and the landmark pixel can be approximated to the color distance, i.e, $g_j - g_k \approx \delta_{jk}$, and then (7) is approximated as

$$\sum_{i \in V} \sum_{k \in Q} \frac{1}{|N_i|} \sum_{j \in N_i} (g_i - g_j - (\delta_{ij}^k))^2 \tag{8}$$

where $\delta_{ij}^k = \delta_{ik} + \delta_{kj}$.

With (8), the modified energy function $\hat{f}(\mathbf{g})$ is finally written as

$$\hat{f}(\mathbf{g}) = (1 - \lambda) \sum_{(i,j) \in E} (g_i - g_j - \delta_{ij})^2 + \lambda \sum_{i \in V} \sum_{k \in Q} \frac{1}{|N_i|} \sum_{j \in N_i} (g_i - g_j - \delta_{ij}^k)^2 \tag{9}$$

and corresponding linear equations are

$$\begin{aligned} (1 - \lambda) \sum_{j \in N_i} (g_i - g_j - \delta_{ij}) + \frac{\lambda}{|N_i|} \sum_{j \in N_i} \sum_{k \in Q} (g_i - g_j - \delta_{ij}^k) \\ + \sum_{j \in N_i} \frac{\lambda}{|N_j|} \sum_{k \in Q} (g_i - g_j - \delta_{ij}^k) = 0, \quad i \in V. \end{aligned} \tag{10}$$

Letting $|N_i| = 4$ for all i , (10) is reduced to

$$\sum_{j \in N_i} (g_i - g_j - \delta'_{ij}) = 0, \quad i \in V \tag{11}$$

where $\delta'_{ij} = ((1 - \lambda)\delta_{ij} + \frac{\lambda}{2} \sum_{k \in Q} \delta_{ij}^k) / ((1 - \lambda) + \frac{\lambda|Q|}{2})$. The main difference of the approximated energy in (9) from the original energy in (2) is that the contribution of landmark pixels to the energy (that measures the global contrast) is incorporated into δ' implicitly and the energy does not rely on landmark pixels any more. Therefore, the linear system induced by (11) becomes symmetric as follows:

$$\mathbf{Sg} = \mathbf{b}' \tag{12}$$

where i th element of \mathbf{b}' is $\sum_{j \in N_i} \delta'_{ij}$. Fortunately, \mathbf{S} is the same as 5-point Laplacian matrix \mathbf{A} except for the elements corresponding to boundary pixels. Hence, we can substitute \mathbf{S} with \mathbf{A} without much change in the solution. In other words, (12) becomes a discrete version of Poisson equation by this substitution, and it is efficiently solved by using fast DCT [11].

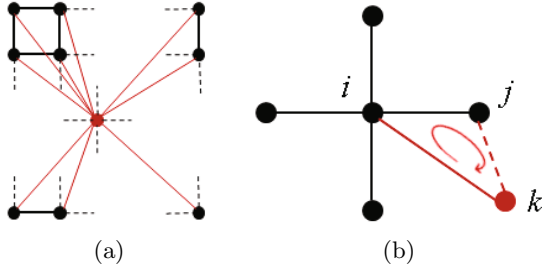


Fig. 4. (a) Graph representation when there is one landmark point. The landmark pixel and Type-G edges are colored red. (b) A loop example. The edge e_{kj} is represented by dashed line, because it is temporarily created when applying zero curl constraint.

Graphical Interpretation of the Energy Approximation. The above energy approximation can easily be understood by the graphical interpretation. The minimization of the proposed energy is actually an analogy of the reconstruction of the grayscale image from the target gradient field δ . However unlike other reconstruction problem such as surface reconstruction [11], the proposed method depends on not only the edges connecting neighboring two pixels (Type-N edge) but also the edges connecting two pixels apart from each other (Type-G edge). Since the existence of Type-G edges makes it impossible to use fast reconstruction method such as 2D Poisson solver, we modify the energy so as to remove Type-G edges from a graph and then modify the target gradient field as a price to pay for removing the Type-G edges.

To graphically illustrate the energy approximation, we define a graph $G(V, E_N, E_G)$ where V is a set of nodes, E_N is a set of all the Type-N edges and E_G is a set of all the Type-G edges. In this example, we select only one pixel as a landmark pixel to simplify the problem. The corresponding graph is illustrated in Fig. 4 (a) where landmark node and Type-G edges are colored red and other nodes and Type-N edges are colored black.

Let us denote the grayscale difference $g_i - g_j$ by g_{ij} . Then, $g_{ik}(k \in Q)$, $g_{kj}(j \in N_i)$ and g_{ji} have to satisfy zero curl constraint [13] as follows :

$$g_{ik} + g_{kj} + g_{ji} = 0. \tag{13}$$

Zero curl constraint states that any closed loop integral is zero on the integrable gradient field and zero curl of each elementary loop guarantees the integrability of gradient field. For example of the elementary loop, we illustrate the loop $(i \rightarrow j \rightarrow k \rightarrow i)$ in Fig. 4 (b). From the zero curl constraint, (7) is easily derived by substituting g_{ik} with $-g_{kj} - g_{ji}$ or $g_{ij} + g_{jk}$. Note that the substitution occurs for all the neighbors and thus the weight $\frac{1}{|N_i|}$ is introduced in (7). Now with an assumption that $g_{ik} \approx \delta_{ik}$, we have (8). Letting the edge connecting i th pixel and j th pixel be e_{ij} , Type-G edge e_{ik} is, after all, removed in (8) and the energy on e_{ik} is distributed to the neighbor edges $e_{ij}, j \in N_i$ to modify the target gradient field. Continuing this for every node, we have three distances at

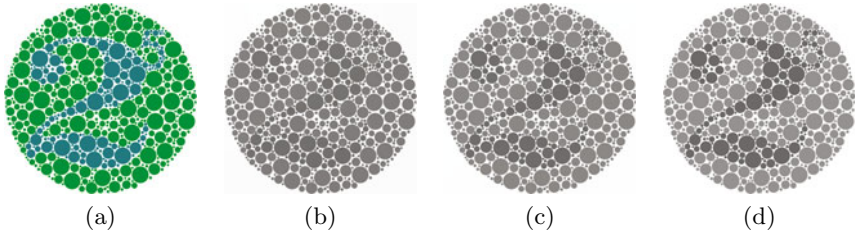


Fig. 5. The experiment for investigating the effect of λ . The contrast is getting strong as λ increases. (a) Original image (b) $\lambda=0$. (c) $\lambda=0.2$ (d) $\lambda=0.9$.

every edge, for example on an edge e_{ij} , δ_{ij} and two δ_{ij}^k s (one is from g_{ik} and the other is from g_{jk} , $j \in N_i$). Combining three distances in the least squares sense with corresponding weights as

$$\tilde{\delta}_{ij} = \operatorname{argmin}(1 - \lambda)(\tilde{\delta}_{ij} - \delta_{ij})^2 + \lambda \frac{2}{4}(\tilde{\delta}_{ij} - \delta_{ij}^k)^2, \quad (14)$$

we have $\tilde{\delta}_{ij} = ((1 - \lambda)\delta_{ij} + \frac{\lambda}{2}\delta_{ij}^k) / ((1 - \lambda) + \frac{\lambda}{2})$. Without loss of generality, this is generalized to the M landmark pixels case and $\tilde{\delta}_{ij}$ is reduced to δ'_{ij} in (11).

As a result, by the energy approximation, the problem is reduced to the reconstruction of the grayscale image from the target gradient field δ' constructed on a new graph $G(V, E_N)$ without dependency on E_G and, in the least squares sense, this is efficiently solved by a fast 2D Poisson solver.

4 Experimental Results

The proposed method is implemented on a PC equipped with Intel Core2Quad 2.4GHz CPU and Agarawal's implementation in [1] is used to solve the 2D Poisson equation with Neumann boundary condition. The current implementation runs on a single core and it takes around 5 seconds in dealing with 1000×800 color image.

We first explore the effect of the control parameter λ in (2). This parameter is for balancing the local contrast with the global contrast, and high value is expected to enhance global contrast. Fig. 5 shows the result of this experiment. The input color image in Fig. 5 (a) is composed of three colors green, blue and white, and white-blue and white-green are in contact, but blue-green is not. As shown in Fig. 5 (b) when $\lambda = 0$, i.e., the global contrast is not considered, the contrast between two colors in contact (white-blue and white-green) are well perceived as experienced in color image, but the contrast between blue and green is very low. On the other hand, contrast gets higher as λ increases as can be seen in Fig. 5 (c) when $\lambda = 0.2$ and Fig. 5 (d) when $\lambda = 0.9$. By this experiment, we can see that the proposed energy formulation is able to balance the local contrast with the global contrast.

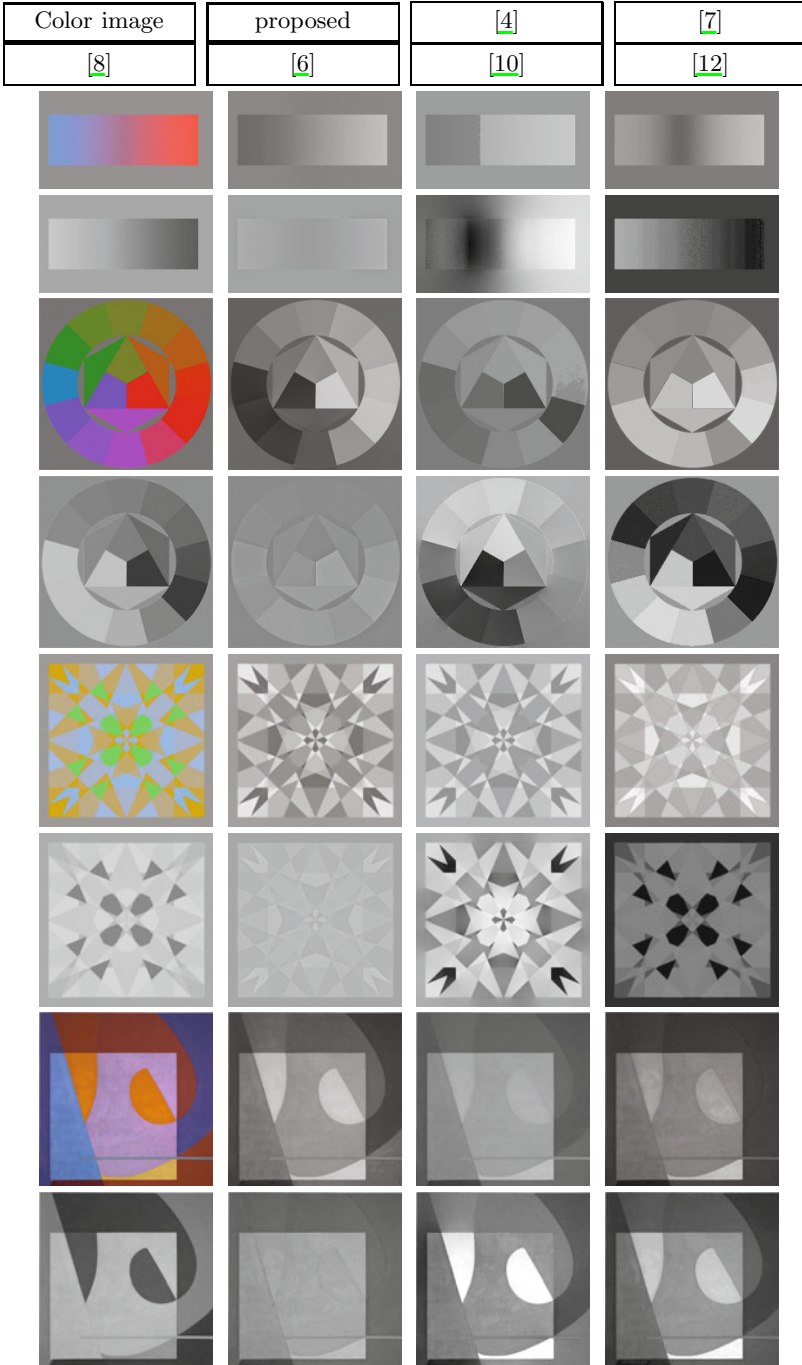


Fig. 6. The experimental results. (Input and result images courtesy of Čadík [2]).

Second, we compare the proposed method to the previous ones on a variety of color images. However in the literature of color conversion, an objective measure has not yet been developed and there is only one trial that compares the existing methods subjectively by Čadík [2]. Hence in this paper, we select several images used in [2] and subjectively compare the proposed method to the existing ones as shown in Fig. 6. The results on the other images are provided as a supplementary file. Throughout all the images, λ is set to 0.3. From the results in Fig. 6, we believe that our method shows better performance than the previous methods. It should be emphasized that the results on the first image (ramp image) shows large variance in performance, that is, the proposed method and the method in [8] give perceptually plausible grayscale image, while the other methods fail. It is because local contrast and global contrast should be simultaneously considered in the case of ramp image and the methods that do so without artifacts are only the proposed method and the method in [8]. Although the methods in [4] and [12] also consider the global contrast, unwanted contours are produced as shown in Fig. 6.

5 Conclusions

We have proposed an energy minimization based method for the perceptually plausible conversion of a color image to a grayscale one. The energy is designed in the way that the contrasts experienced in the color image are preserved in the grayscale image as well and thus perceptually accurate grayscale image is obtained. We have also presented a fast implementation by approximating the energy function. The minimizing the approximated energy is the same as reconstructing the grayscale image from given gradient field over standard 4-neighborhood system and this is efficiently solved by using the fast 2D Poisson solver.

Acknowledgement. This research was performed for the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Knowledge Economy (MKE).

References

1. Agrawal, A., Raskar, R.: What is the range of surface reconstructions from a gradient field. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 578–591. Springer, Heidelberg (2006)
2. Čadík, M.: Perceptual evaluation of color-to-grayscale image conversions. *Pacific Graphics* 27, 1745–1754 (2008)
3. Davis, T.A.: *Direct Methods for Sparse Linear Systems*. SIAM, Philadelphia (2006)
4. Gooch, A.A., Olsen, S.C., Tumblin, J., Gooch, B.: Color2gray: salience-preserving color removal. In: SIGGRAPH, vol. 24, pp. 634–639 (2005)
5. Heckbert, P.: Color image quantization for frame buffer display. *Computer Graphics* 16, 297–307 (1982)

6. Kaleigh, S., Pierre-Edouard, L., Joëlle, T., Karol, M.: Apparent greyscale: A simple and fast conversion to perceptually accurate images and video. In: Eurographics, vol. 27 (2008)
7. Kim, Y., Jang, C., Demouth, J., Lee, S.: Robust color-to-gray via nonlinear global mapping. In: SIGGRAPH ASIA, vol. 28 (2009)
8. Mark, G., Dodgson, N.A.: Fast, contrast enhancing, color to grayscale conversion. *Pattern Recogn.* 40, 2891–2896 (2007)
9. Nayatani, Y.: Simple estimation methods for the helmholtz-kohlrausch effect. *Color Res. Appl.* 22, 385–401 (1997)
10. Neumann, L., Čadík, M., Nemcsics, A.: An efficient perception-based adaptive color to gray transformation. In: *Computational Aesthetics in Graphics, Visualization, and Imaging*, pp. 73–80 (2007)
11. Press, W., Teukolsky, S., Vetterling, W., Flannery, B.: *Numerical Recipes in C*. Cambridge University Press, Cambridge (1992)
12. Rasche, K., Geist, R., Westall, J.: Re-coloring images for gamuts of lower dimension. In: Eurographics, vol. 24, pp. 423–432 (2005)
13. Rama, A.A., Agrawal, A.: An algebraic approach to surface reconstruction from gradient fields. In: ICCV, pp. 174–181 (2005)
14. Saad, Y.: *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia (2003)

Affordance Mining: Forming Perception through Action

Liam Ellis^{1,2}, Michael Felsberg¹, and Richard Bowden²

¹ CVL, Linköping University, Linköping, Sweden

² CVSSP, University of Surrey, Guildford, UK

Abstract. This work employs data mining algorithms to discover visual entities that are strongly associated to autonomously discovered modes of action, in an embodied agent. Mappings are learnt from these perceptual entities, onto the agents action space. In general, low dimensional action spaces are better suited to unsupervised learning than high dimensional percept spaces, allowing for structure to be discovered in the action space, and used to organise the perceptual space. Local feature configurations that are strongly associated to a particular ‘type’ of action (and not all other action types) are considered likely to be relevant in eliciting that action type. By learning mappings from these relevant features onto the action space, the system is able to respond in real time to novel visual stimuli. The proposed approach is demonstrated on an autonomous navigation task, and the system is shown to identify the relevant visual entities to the task and to generate appropriate responses.

1 Introduction

This paper proposes a method for discovering the visual features that are important to a vision system given a specific problem (e.g. a robotics tasks). This is achieved by first applying unsupervised learning in the problem output space (e.g. the agent’s actions). The structure discovered in the output space is then used to organise the input space (e.g. the agent’s perceptual representation), in order to form meaningful input representations. This organisation process is achieved by finding strong associations between modes of the output space and configurations of the input space. Association rule data mining algorithms are employed to efficiently find these associations.

This work is motivated by a desire for adaptive cognitive vision systems, that build their own visual representations based on experience and learn how to react to their environment, without the need for explicit definitions of representations or strategies by an engineer. Such emergent systems should be less ‘brittle’ than conventional hard-coded systems, and demonstrate increased robustness when faced with changes in the environment not envisaged by the engineer.

In natural cognitive systems, increased sensory complexity, along with the machinery used to interpret such complexity, is generally associated with an increasing ability to interact with and manipulate the environment, facilitated by increasing motor capabilities. It is straightforward to see that the complexity of

interaction a system can demonstrate - its motor capabilities - is to a certain extent determined by the complexity of its perceptual system. It is, perhaps, less straightforward to see that the complexity of a systems perceptual system, is determined by the complexity of the systems motor capabilities. However, this apparent cyclical causality, linking perceptual and motor capabilities is supported by a significant body of work in modern cognitive sciences, and has firm philosophical [1] and neurophysiological [2] foundations. In particular the theory of *embodiment*, a term used within psychology, philosophy, robotics and artificial intelligence, is based on the premise that the nature of the mind is determined by the embodiment of the cognitive agent [1] [3]. Related to this is *affordance* theory, that states that the world is perceived not only in terms of object shapes and spatial relationships but also in terms of object possibilities for action [4]. The work presented here demonstrates an embodied approach to constructing an affordance based representation of the world.

Data mining algorithms are useful for efficiently identifying correlations in large symbolic datasets. These methods have begun to be applied to vision tasks such as: identifying features which have high probability of lying on previously unseen instances of an object class [5], mining dense spatio-temporal features for multi-action recognition [6], and finding near duplicate images within a database of photographs [7]. These methods benefit from both the scalability and the efficiency of data mining methods. This work employs data mining algorithms to the novel domain of percept-action association mining. The mechanism of mining frequent and distinctive feature configurations employed here is most similar to that of Quack et al. [5], however, here the discovered configurations are used directly in an action generation process, rather than as a pre-processing step for identifying useful features for other classification techniques. Furthermore, whilst in [5] supervision is required to label the classes of objects that are learnt, in this work, classes of actions are obtained by an unsupervised learning approach.

The rest of this paper is organised as follows: In section [1.1] background to association rule mining is presented. In section [1.2] the robotic platform, training method and intended task are briefly detailed. Section [2] describes the central mechanism of action space clustering and how this identifies classes of actions and percept groupings. Section [3] presents a complete overview of the proposed system, identifying the key processing stages involved, which are presented in detail in sections [4] and [5]. Section [4.1] details the approach used to encode visual information as feature configurations and section [4.2] presents the method for finding associations between classes of actions and these feature configurations. Section [5] details how mappings are learnt between associated percept and action data and how these mappings are exploited to generate responses to novel image data. Section [6] presents the experimental evaluation of the system and section [7] contains a discussion and conclusions.

1.1 Association Rule Mining

Association rule mining is the process of finding association rules in a database $D = \{t_1, t_2, \dots, t_m\}$ of transactions, where each transaction is a set of items, and

I is the set of all items¹. An association rule is an implication of the form $X \Rightarrow Y$ where $X, Y \subseteq I$ and $X \cap Y = \emptyset$.

Association rules are selected from the set of all possible rules based on constraints on measures of significance and interest. These constraints are thresholds on itemset *support* and rule *confidence*. The support, $supp(X)$, of an itemset X is defined as the proportion of transactions in the database which contain X . The confidence, $conf(X \Rightarrow Y)$ of a rule is defined:

$$conf(X \Rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)} \quad (1)$$

The Apriori algorithm [8] employed in this work, exploits the anti-monotonicity of the support threshold constraint - that a subset of a frequent itemset must also be a frequent itemset - to efficiently mine association rules. This work uses an efficient existing implementation of the Apriori algorithm [9].

1.2 Robotic Platform and Training Data Collection

The robotic platform developed is a relatively inexpensive platform for the investigation of embodied artificial cognitive agents. Based on a standard Remote Control (RC) model car fitted with a wireless camera, the system allows a teacher to demonstrate the desired driving behaviour by viewing the images from the camera on a PC monitor and using a standard computer game steering wheel and foot pedal controller to navigate the car.²

The training process involves the teacher driving the agent in order to follow a lead vehicle. This collects a sequence of pairs of images and control parameters that implicitly capture the desired behaviour.

2 Action Space Clustering

Unsupervised learning techniques are often applied to percept spaces (e.g. image or feature space), but are prone to yielding ambiguous or erroneous results. This is often due to assumptions about suitable distance metrics used to cluster the data. In general, action data (e.g. control signals) are of lower dimensionality than percept data, and related points in the action domain are generally more similar than related points in the percept domain [10]. This implies that the action space is more suited to unsupervised learning techniques. These observations lead to the proposition that the action space should drive the organisation of the percept space. This idea is strongly related to embodiment, and the Embodied Mind theory [1] [3].

For an embodied agent (e.g. all natural cognitive systems and the system proposed in this work), percept data is never obtained in isolation - it is always

¹ The terminology *transactions* and *items* comes from the data mining literature, reflecting the subjects origins in market basket analysis applications.

² Details of robotic platform and collected data sets and code available here www.cvl.isy.liu.se/research/embodied-vehicle-navigation

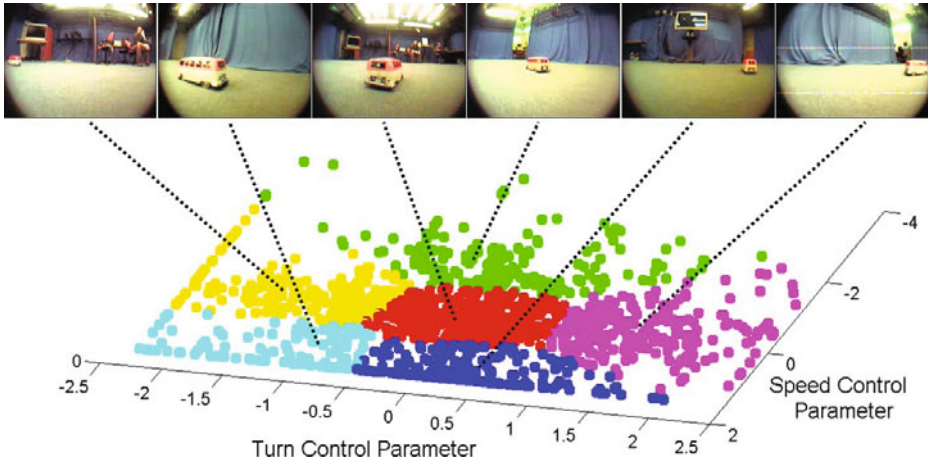


Fig. 1. *Action clustering:* Action clusters are formed along with sets of associated images

coupled to action data. This coupling is exploited in this work by clustering coupled percept-action exemplars, in the action space. This results in the formation of meaningful classes of action or ‘action-types’, as well as meaningful perceptual groups. The action data, $\{\mathbf{a}^1 \dots \mathbf{a}^N\}$, with $\mathbf{a}^n = [a_{turn}^n, a_{speed}^n] \in \mathbb{R}^2$, is clustered - using k-means clustering - into $k_{act} = 6$ clusters. Figure 1 illustrates the result of performing this action space clustering and examples of the associated images are shown. In order to obtain invariance to displacement, scale and rotation, the action data is whitened prior to clustering. The data is translated (by the mean sample value), scaled (each dimension by the associated eigen values of the sample covariance matrix) and rotated such that the features have zero mean, unit variance and the data axis coincide with the eigenvectors of the sample covariance matrix.

3 System Overview

An overview of the proposed approach is illustrated in figure 2. First an exemplar set, E , of training data of the form $E = \{(\mathbf{p}^1, \mathbf{a}^1), \dots, (\mathbf{p}^N, \mathbf{a}^N)\}$, where $\{\mathbf{p}^1 \dots \mathbf{p}^N\}$ is the set of images, and $\{\mathbf{a}^1 \dots \mathbf{a}^N\}$ is the set of action vectors, is collected (details of this training process are given below). Symbolic representations of both the actions, and percepts, are then formed. For the action data, k-means is applied directly to action vectors, resulting in k_{act} action-types, as detailed above. For image data, a visual codebook of SIFT features is built using k-means clustering, where the cluster centers make up the codewords. Spatial relationships between features are represented by encoding local feature configurations, as described in section 4.1. The visual information in each image is thus represented as a set of codeword configurations.

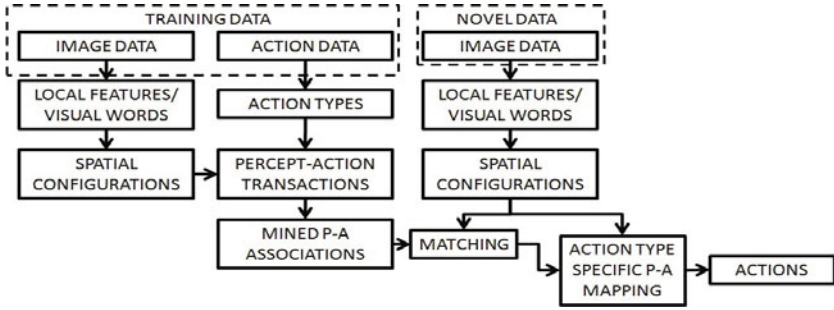


Fig. 2. *System overview:* Coupled percept and action data are represented as Percept-Action (P-A) transaction vectors by concatenating visual codeword configuration vectors and action-type labels. Data mining is then used to discover P-A associations that identify feature configurations that are associated to a particular action-type. Matching these association rules in training images then provides data for learning P-A mappings for each association rule, that map from feature configurations to actions. Matching the association rules in novel images then activates the associated P-A mappings, thus providing a mechanism for generating appropriate responses to novel image data.

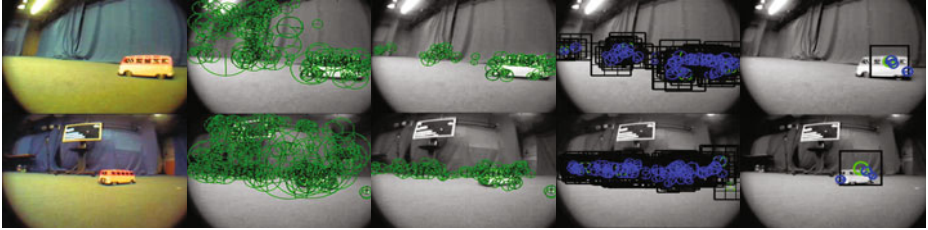
Links between the symbolic percept and action spaces are then obtained by performing data mining on a combined Percept-Action (P-A) representation, named P-A transactions. Each transaction represents an action-type coupled to a codeword configuration, where one item in each transaction represents the action-type, and the remaining items represent a visual codeword feature configuration, as detailed in section 4.2. The data mining algorithm then processes these transactions to produce P-A association rules.

The training data, and the mined association rules are then used to learn action-type specific P-A mappings, as in section 5.1. These mappings map from the continuous (un-quantised) pose of the image features associated to an action-type, onto the continuous action vectors belonging to that action-type. These mappings constitute affordances for the mined perceptual entities.

Still referring to figure 2, when presented with novel image data, the system constructs the visual codeword configurations as before. These configurations are matched to the mined association rules and the P-A mappings associated to the rules are applied to the features that form the matching configurations, in order to generate a response. This process of generating responses to novel image data is detailed in section 5.2.

4 Mining Percept-Action Associations

The proposed vision system is based on local feature descriptors. A Difference of Gaussian (DoG) detector is used to extract regions and the SIFT descriptor [11] is used to describe the regions. A prior is placed on the scale and location of the SIFT features used in the later stages of the process. This results in a filtering of the set of SIFT descriptors extracted from each image. Figures 3b and 3c



(a) Input image. (b) Sift descriptors. (c) Sift filtering. (d) Feature configurations. (e) Mined configurations.

Fig. 3. *P-A mining process*: Five stages of the feature mining process are illustrated. Sift descriptors are extracted from the input images. These are then filtered to remove features near the top of the image or that have overly large scales. Feature configurations are then assembled and those configurations that are associated to particular action-type are then discovered through data mining.

illustrate this filtering stage. As the lead vehicle will always remain on the ground plain, and as features on the lead vehicle will have a limited scale in the images, features are rejected that appear too near the top of an image or have overly large scales.

The 128-dimensional SIFT feature descriptors are clustered to form a visual word vocabulary, using k-means clustering. Additionally, the scale and orientation of the features are clustered to form ‘scale words’ and ‘orientation words’. Meaning that each SIFT feature can be described using three discrete labels - descriptor, scale and orientation words - and the continuous horizontal and vertical position. For clustering the descriptor, $k_{desc} = 50$, for scale and orientation, $k_{scale} = 5$, $k_{orient} = 5$.

4.1 Feature Configurations

Figure 4 illustrates the method used to encode the spatial configuration of the extracted SIFT features. A similar scheme was introduced in [5]. For every feature in an image (after filtering) a 3-by-3 grid is placed on the image, centered on the feature, and scaled proportionally to the feature scale. Any neighbouring features that fall into a tile of the grid are encoded as part of that feature configuration, the encoding reflects which tile the feature is in i.e. it’s spatial relation, and the visual, scale and orientation words representing the feature. A sparse vector representation is employed for which the non-zero indices encode the configuration and the values store the feature index in the image, so that the continuous feature pose may be recalled for the P-A mappings. The feature configuration vector contains the indices of the non-zero elements of the sparse vector, and is used to represent the visual information in the data mining process.

Examples of feature configurations for two of the training images are shown in figure 5. As can be seen, some of the feature configurations lie on or partially

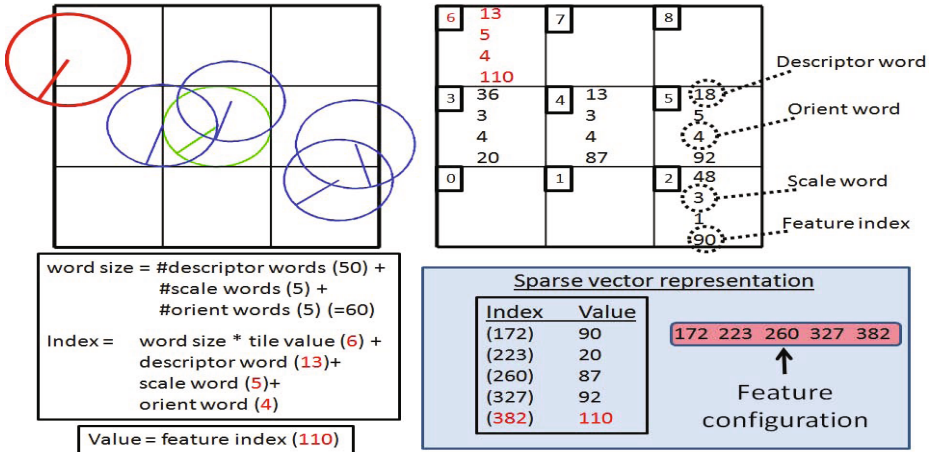


Fig. 4. Encoding configurations: This figure illustrates how a configuration of features is encoded in a sparse vector representation, and how this sparse vector representation is used to build the feature configuration vectors used by the mining algorithm. The top left of the figure shows a configuration of features found around the central (green) feature. The top right of the figure illustrates how the feature configuration is represented as a configuration of visual codewords at quantised relative locations, scales and orientations. The bottom left part of the figure details how a particular feature (marked in red in the top left) is encoded in the sparse vector representation. The bottom right of the figure shows the sparse vector representation of the configuration. Also shown is the feature configuration vector that forms the percept part of the transaction vectors used in the data mining. The values of the non-zero indices of the sparse vectors are the feature indices that identify the feature in the image, these are used when mapping from feature pose to action parameters. Note that the center feature (green) is not represented.

on the target vehicle, whilst many lie on the background. The full set of configurations for an image (as illustrated in figure 3d) will contain considerable redundancy, where each local pairwise spacial relationship will be encoded a number of times within multiple feature configurations.

4.2 Percept-Action Transaction Database

A Percept-Action (P-A) transaction represents a feature configuration coupled to the associated action-type. The action-type being the cluster label assigned to the action parameters that are associated to the image from which the feature configuration is extracted.

The set of items is $I = \{\alpha_1, \dots, \alpha_k, R_1, \dots, R_l\}$, where $\{\alpha_1, \dots, \alpha_k\}$ are the $k = 6$ action-type items and $\{R_1, \dots, R_l\}$ are the $l = 540$ (9 tiles, 50 visual, 5 orientation and 5 scale words) unique spatial relationships that form the feature configurations. Each transaction vector is the concatenation of the action-type item

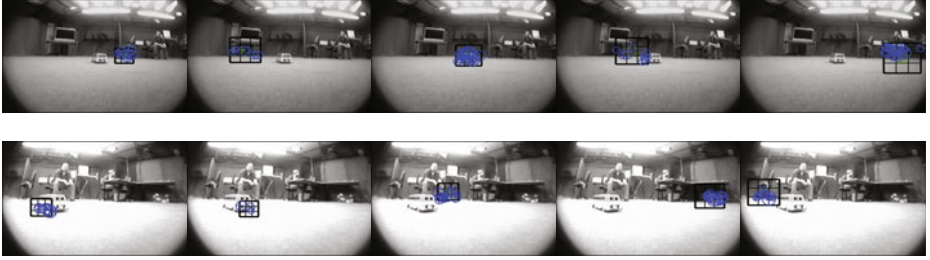


Fig. 5. *Feature configurations:* Five examples of feature configurations for two frames are shown. Some of the configurations contain features on the target, some contain features only from the background.

with the items from the feature configuration vector, as illustrated in figure 6. Therefore each transaction contains a subset of I with one item always drawn from $\{\alpha_1, \dots, \alpha_k\}$.

The transaction database $D = \{t_1, t_2, \dots, t_m\}$ is assembled, as in figure 6, by collecting together all P-A transactions drawn from all training data, $E = \{(\mathbf{p}^1, \mathbf{a}^1), \dots, (\mathbf{p}^N, \mathbf{a}^N)\}$. In the experiments carried out in section 6, the total number of transactions in the database, $m = 88810$. This database is then processed using the Apriori [9] data mining algorithm, in order to find frequent and discriminative feature configurations for each action-type.

4.3 Mining P-A Association Rules

Association rule mining is employed to mine the P-A transaction database, in order to discover feature configurations that frequently co-occur with a particular action-type, and not all other action-types. The algorithm finds subsets of items from the transaction vectors that are frequent and discriminative to a given action-type. The Apriori algorithm is run once for each action-type, where it searches for rules including that action-type, and treats all other action-types as negative examples.

For the experiments carried out here, the support threshold $T_{Supp} = 0.02$ and confidence threshold $T_{Conf} = 99$ are used for all action-types and are selected by experimentation. These values are chosen as they provide an appropriate size set of rules to allow for real time rule matching in novel images (as detailed below in section 5.2). Between 400 and 500 rules are found for each action-type. The rules contain between 3 and 10 items (including the action-type item). An example of such a rule would be $\{slow-left \rightarrow 114, 188, 295\}$, meaning that a particular configuration of three features has been associated with actions of the type ‘slow-left’.

For the mining, the feature configurations are represented using the indices of the non-negative elements of the sparse vector representation, as illustrated in figures 4 and 6. However, when matching configurations found in an image

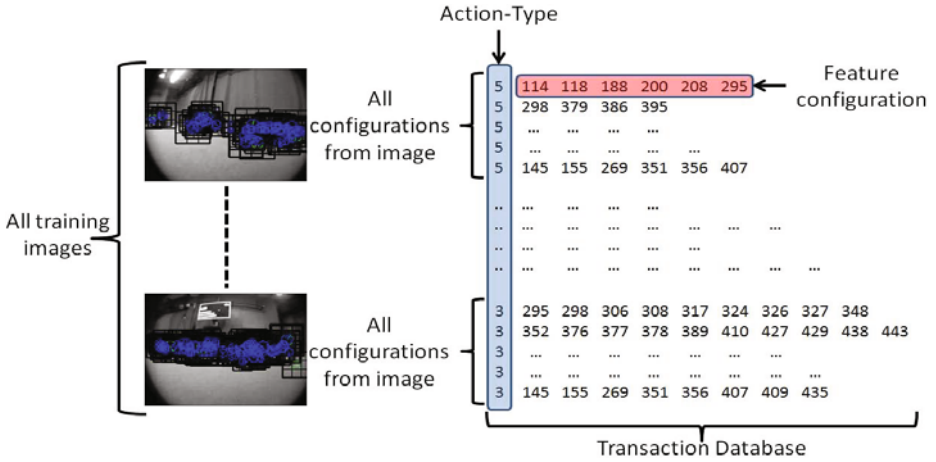


Fig. 6. *Transaction database:* Each transaction is the concatenation of an action-type label (obtained by k-means clustering the action parameters) with a feature configuration (the indices of the non-zero elements of the sparse vector representation). The transaction database is the collection of all transactions from all training images.

to association rules, the sparse vector representation is used. The dot product is used to efficiently match rules to configurations found in an image. Examples of the mined association rules for each action-type are illustrated in figure 7

5 Affordance Based Representation

This section details how the proposed system builds an affordance based representation of the world, and how this representation is used to generate responses to novel percept data. This is achieved by attaching learnt mappings to each mined association rule. These map from the pose (horizontal and vertical position, scale and orientation) of the features in rules onto actions. Linear regression is used to learn linear mappings from pose space to action space.

5.1 Learning Action-Type Specific P-A Mappings

A linear percept-to-action (P-A) mapping, \mathbf{H}_{P-A} , is learnt for each association rule (mined configuration). \mathbf{H}_{P-A} maps from $(C * 4)$ -dimensional feature pose space, to 2-dimensional action space, $\mathbb{R}^{C*4} \rightarrow \mathbb{R}^2$, where C is the number of features that make up the rule. A bias term is included in the linear model. An action, \mathbf{a} , is computed from a $(C * 4)$ -dimensional pose vector, $\hat{\mathbf{p}}$, as in equation 2

$$\mathbf{a} = \mathbf{H}_{P-A}\hat{\mathbf{p}} + b \tag{2}$$

In order to learn each \mathbf{H}_{P-A} , N training examples of $\{\mathbf{a}_i, \hat{\mathbf{p}}_i\}$ pairs, $(i \in [1, N])$ are required. The training set for each \mathbf{H}_{P-A} is obtained by matching rules to

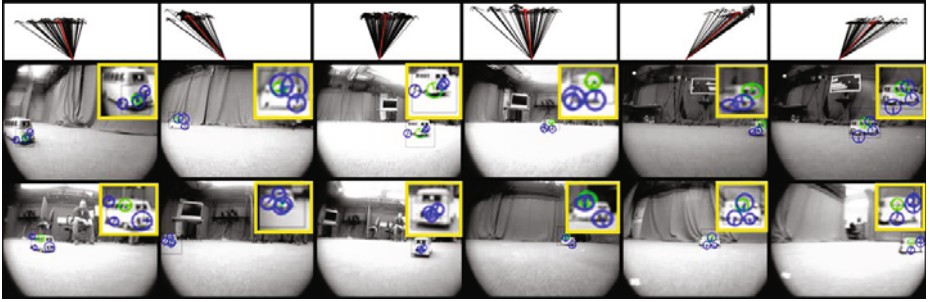


Fig. 7. *Association rules:* Training vectors for six action-types (from left to right on top row: ‘slow-left’, ‘fast-left’, ‘slow-straight’, ‘fast-straight’, ‘fast-right’, ‘slow-right’) are shown along with examples of associated configuration rules mined for each type. In general, if the lead vehicle is to the left/center/right, then the associated action is left/center/right. However sometimes the pose of the lead vehicle, rather than the position is used to associate to the action-type (e.g. far right on middle row, and second from right bottom row).

configurations found in the training images. Whenever a configuration found in a training image is matched to a rule, the pose parameters of the features that make up that configuration form a new pose vector $\hat{\mathbf{p}}$. The value of the non-negative elements of the sparse vector provide the index to the matched configurations constituent features.

For each rule, all the matched configuration pose vectors, $\hat{\mathbf{p}}$, and the associated action vectors, \mathbf{a} , are stacked into the training matrices, \mathbf{P} and \mathbf{A} respectively. To learn the bias for the linear model an additional column of 1s is added to the end of \mathbf{P} , giving: $\mathbf{P}' = (\mathbf{P}, [1])$, where $[1]$ denotes a column vector of N rows. Using least squares, \mathbf{H}_{P-A} can now be obtained as follows:

$$\mathbf{H}_{P-A} = \mathbf{A}\mathbf{P}'^+ = \mathbf{A}\mathbf{P}'^T(\mathbf{P}'\mathbf{P}'^T)^{-1} \quad (3)$$

Where \mathbf{P}'^+ is the pseudo inverse of \mathbf{P}' .

5.2 Responding to Novel Data

A new input image is processed to generate a set of visual codeword feature configurations as detailed above. Configurations are then compared to all the mined action-type specific configurations (rules). Matching a configuration to a mined rule is achieved by computing the dot product of the two sparse vector representations. If the number of non-zero elements in the dot product is equal to the number of non-zero elements in the sparse vector representation of the association rule, then the rule is matched. If a match is found then an action prediction is made as in equation 2 using the \mathbf{H}_{P-A} associated to the matched rule. Once all found configurations have been compared to all rules, the output action is computed as the median of all action predictions.

To speed up the generation of actions, only configurations within a search range of the previous target location are compared to the rules. The search range is proportional to median grid size of the configurations matched in the previous frame, and is centered at the median position of the previously matched configurations.

6 Evaluation

The two objectives of this paper - to discover the visual entities important to the task and to generate appropriate responses to novel data - are evaluated. This is achieved by using ground truth data for the target vehicle position. This data is obtained by learning (in a supervised manner) a detector for the lead vehicle. The detector is a Waldboost detector [12] trained on hand labeled examples - sufficient examples are used in training to provide a detector that achieves very high accuracy on the test dataset. The position of the lead vehicle is then used to evaluate how well the mined configurations relate to the lead vehicle. Additionally, the ground truth data is used as input to a supervised method for action generation, to compare to the proposed unsupervised approach.

Table 1. Hit/miss ratio for mined configurations lying on the lead vehicle

Action class	slow-left	fast-left	slow-straight	fast-straight	fast-right	slow-right
Hit/Miss ratio	0.95	0.78	0.83	0.74	0.92	0.87

Figure 7 shows examples of mined configurations that lie on the object of interest, the lead vehicle. Indeed the majority of mined configurations do lie on the lead vehicle, implying that the proposed method has discovered the important visual entities. To quantitatively evaluate this, the hit/miss ratio is measured across a test set of unseen data. A hit is defined as when at least 50% of the features that make up a configuration lie within the bounding box obtained from the detector. Table 1 shows the hit/miss ratio for each action-type.

The action generation mechanism is evaluated by comparing the actions generated by the system on unseen test data with actions generated by a supervised approach. The supervised approach maps from the ground truth target pose to the action parameters using a single linear regression model, the same as in the proposed approach. In figure 8 it can be seen that the signals generated by both the approaches approximately follow the expected signals.

Comparing the action signals generated by the supervised and proposed (unsupervised) approaches (figure 8), it can be seen that both methods approximately reproduce the control signal provided by the teacher. Note that the high accuracy of the supervised approach in parts of the signal, reflects the strongly linear relationship between target pose and action signals.

The large peaks in the signal generated by the supervised approach correspond to false detections. Although there are false detections (incorrect configuration matches) in the proposed system, these generally have a minimal effect on the

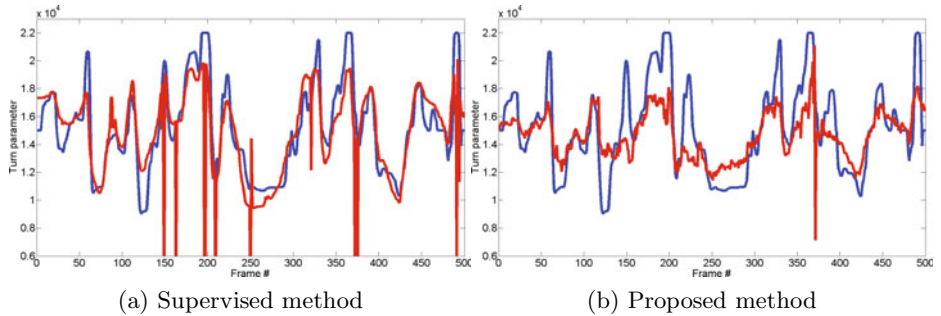


Fig. 8. *Generated action signals:* The generated ‘turn-control’ action signals (red) are shown for the proposed method and a supervised method, along with the expected action signal (blue)

output as the output is the median of a number of predictions, therefore these irregularities in the action signals are generally avoided.

Certain parts of the signal generated by the proposed approach do not exactly follow the expected signal (for example from frame 100 to 150). This is in some cases due to the fact that the expected signal, provided by the teacher, includes instances of oversteer and compensation, and is therefore not necessarily superior to the generated signal.

Figures 9 and 10 demonstrate the approach at imitating the desired behaviour. In figure 9 the target is placed at three stationary positions and the agent is shown to generate actions that drive toward the target. In figure 10 the lead vehicle is driven around and the agent is shown demonstrating the desired behaviour - following the lead vehicle.

7 Discussion

This work presents a method for discovering the visual entities that are important to a given autonomous navigation task and utilising these perceptual representations to imitate the behaviour that is demonstrated by the teacher. The system requires no explicit definition of behaviour, uses no prior model of the objects of interest to the task and no supervision, other than the provision of input-output exemplars in the form of images and actions i.e. recorded experiences that exhibit the desired behaviour.

Partitioning the training exemplars using similarity of actions provides a means of organising the perceptual space of the agent in a way that is relevant to the problem domain. This allows for the discovery of perceptual representations that are specific to a particular class of actions. These representations are discovered using efficient association rule mining techniques. The representations are built on a spatially encoded visual word representation. The results shown in figure 7 and table 1 confirm that the visual entities discovered do in fact relate to the object in the scene that is important to the task.

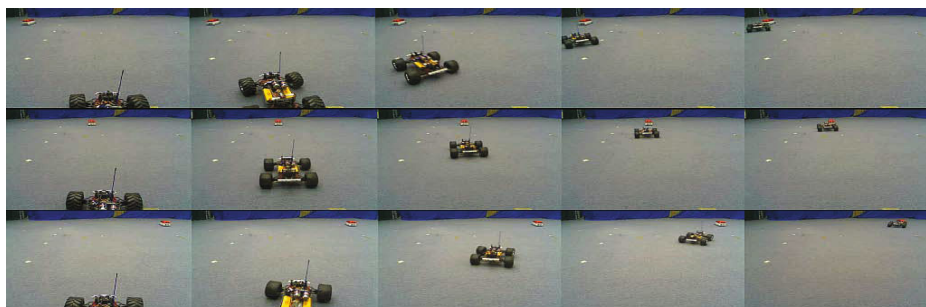


Fig. 9. *Action generation results:* The agent is shown to demonstrate the appropriate actions, by driving (to left - top, straight - middle, to right - bottom) toward the target and then coming to stop

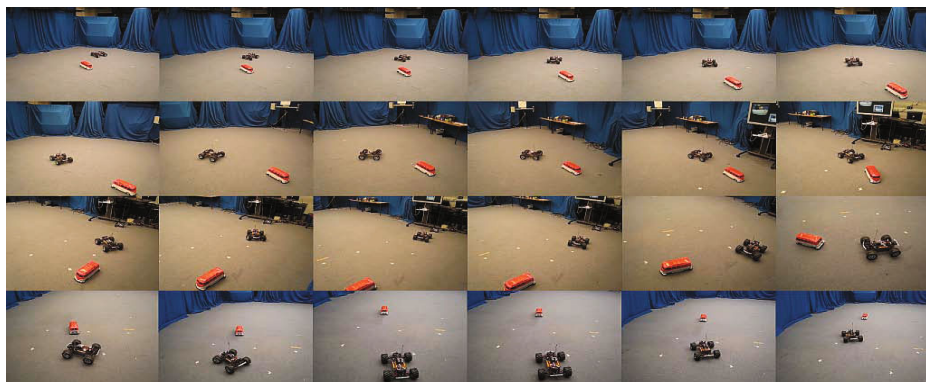


Fig. 10. *Behaviour imitation:* The behaviour demonstrated by example is replicated by the agent, as it follows the lead vehicle

By attaching action generation models (linear percept-to-action mappings) to each discovered visual entity, the system builds an affordance based representation of the world. This novel representation directly couples percepts to actions, resulting in a system that is able to respond to novel percepts in real time. The results presented in figures 8, 9 and 10 demonstrate that this novel affordance based representation generates the type of actions expected and allows the system to imitate the behaviour demonstrated by the teacher, when presented with new situations. This is achieved with no explicit definition of the behaviour.

Choosing $k_{act} = 6$ ensures that there is sufficient inter and intra class variance of visual information whilst also ensuring sufficient exemplars for learning the visual representations and mappings for each action-type. Larger k_{act} reduces the number of training examples for both the configuration mining and mapping learning. Smaller k_{act} increases within class variation and reduces the discriminative power of the mined configurations. Clearly the selection of k_{act} will impact on the quality of both the mined configurations and the generated actions. Future

work will investigate the effect of this parameter on system performance, and investigate the use of mode seeking and other clustering algorithms for action space clustering.

Acknowledgement. This research has received funding from the EC's 7th Framework Programme (FP7/2007-2013), grant agreements 21578 (DIPLECS) and 247947 (GARNICS).

References

1. Lakoff, G., Johnson, M.: *Philosophy in the Flesh: The Embodied Mind and Its Challenge to Western Thought*. Basic Books, New York (1999)
2. Garbarini, F., Adenzato, M.: At the root of embodied cognition: Cognitive science meets neurophysiology. *Brain and Cognition* 56, 100–106 (2004)
3. Brooks, R.A.: Intelligence without reason. In: Myopoulos, J., Reiter, R. (eds.) *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI 1991)*, pp. 569–595. Morgan Kaufmann publishers Inc., San Mateo (1991)
4. Gibson, J.J.: *The Theory of Affordances*. Lawrence Erlbaum, Mahwah (1977)
5. Efficient Mining of Frequent and Distinctive Feature Configurations. In: *IEEE 11th International Conference on Computer Vision, ICCV 2007* (2007)
6. Gilbert, A., Illingworth, J., Bowden, R.: Fast realistic multi-action recognition using mined dense spatio-temporal features. In: *Proc. Int. Conference Computer Vision, ICCV 2009* (2009)
7. Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A.: Total recall: Automatic query expansion with a generative feature model for object retrieval. In: *Proceedings of the 11th International Conference on Computer Vision, Rio de Janeiro, Brazil* (2007)
8. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: *Proceedings of the 20th International Conference on Very Large Data Bases, VLDB 1994*, pp. 487–499. Morgan Kaufmann, San Francisco (1994)
9. Borgelt, C.: Efficient implementations of apriori and eclat (2003)
10. Granlund, G.H.: The complexity of vision. *Signal Processing* 74, 101–126 (1999) (invited paper)
11. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60, 91–110 (2004)
12. 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)*, vol. 2, pp. 150–156. IEEE Computer Society, Washington (2005)

Spatiotemporal Contour Grouping Using Abstract Part Models

Pablo Sala¹, Diego Macrini², and Sven Dickinson¹

¹ University of Toronto

² Queen's University

Abstract. In recent work [1], we introduced a framework for model-based perceptual grouping and shape abstraction using a vocabulary of simple part shapes. Given a user-defined vocabulary of simple abstract parts, the framework grouped image contours whose abstract shape was consistent with one of the part models. While the results showed promise, the representational gap between the actual image contours that make up an exemplar shape and the contours that make up an abstract part model is significant, and an abstraction of a group of image contours may be consistent with more than one part model; therefore, while recall of ground-truth parts was good, precision was poor. In this paper, we address the precision problem by moving the camera and exploiting spatiotemporal constraints in the grouping process. We introduce a novel probabilistic, graph-theoretic formulation of the problem, in which the spatiotemporal consistency of a perceptual group under camera motion is learned from a set of training sequences. In a set of comprehensive experiments, we demonstrate (not surprisingly) how a spatiotemporal framework for part-based perceptual grouping significantly outperforms a static image version.

1 Introduction

Interest in the perceptual grouping of image contours peaked in the late 1990's, when the mainstream object recognition community was primarily shape-based and the bottom-up recovery of distinctive indexing structures was critical in identifying a small number of candidate objects (from a large database) present in the scene. However, the advent of appearance-based recognition (and a corresponding movement away from shape), combined with the reformulation of the recognition problem as a detection problem (in which the image is searched for a single target object), diminished the role of perceptual grouping. Even with the re-emergence of image contours as the basis for categorical models (e.g., [2]), the continuing focus on object detection means that the stronger shape prior offered by a detector subsumes the domain-independent shape priors that make up the non-accidental properties that define perceptual grouping. In other words, the process of domain-independent, bottom-up perceptual grouping to extract a meaningful indexing structure in order to select promising candidates is unnecessary, since in a detection task we know what, i.e., which candidate, we're looking for.

There are clear signs that the community is moving back toward unexpected object recognition, i.e., identifying an image of an unknown object from a large database. Since a linear search through a space of object detectors clearly does not scale to large databases, we must drastically prune the space of candidate detectors to apply to the image. This, in turn, means recovering distinctive image structures that can effect such pruning – a return to perceptual grouping. Yet a simple return to classical grouping techniques is insufficient, for while non-accidentally related contours in an image may be grouped, there is still a semantic gap between the resulting contour groups and the shape structures that comprise a categorical shape model. Only when the contour groups are *abstracted* can they be matched to categorical models.

In recent work [1], we developed a framework in which a small vocabulary of abstract part shape models were used to both group and abstract image contours, yielding a covering of the image with a set of 2-D abstract parts which model the projections of the surfaces of a set of abstract volumetric parts that describe the coarse shape of the object. Thus, rather than invoking an object-level shape prior (detector), which we don't have since we don't know what we're looking at, we instead invoke a small, finite set of intermediate-level, domain-independent shape priors to drive the grouping and abstraction processes (we assume only that the parts can be assembled to describe a significant portion of any object in the database). While the method shows clear promise, there is a fundamental trade-off between abstraction and ambiguity; as a greater degree of abstraction of a set of image contours is allowed, the more ambiguous the abstraction, i.e., the abstraction is consistent with an increasing number of shape models.

In this paper, we exploit the dimension of temporal coherence to help cope with the ambiguity of a shape abstraction inherent in a single static image. Like in [1], we rely on a small, user-defined, abstract shape vocabulary to drive the process of perceptual grouping in a single frame. However, unlike [1], which restricts its analysis to a single image, we assume access to a video sequence in which there is relative motion between the camera and the object, and exploit the spatiotemporal coherence of a perceptual group to reduce false positives that are abundant in a single image. If a perceptual group of contours is consistent with an abstract part model, and is stable over time in terms of its shape (continues to match the same part model) and pose, then we consider the perceptual group to be non-accidental. We introduce a novel probabilistic, graph-theoretic formulation of the problem, in which the spatiotemporal consistency of a perceptual group under camera motion is learned from a set of training sequences. In a set of comprehensive experiments, we demonstrate (not surprisingly) how a spatiotemporal framework for part-based perceptual grouping significantly outperforms a static image version.

2 Related Work

The problem of using simple shape models to group and regularize 2-D contour data has been extensively studied in the past. Many have approached this problem assuming figure-ground segmentation, i.e., they take as input a silhouette,

while others have assumed knowledge of the object present in the scene, i.e., object-level shape priors. In our approach, we assume neither; rather, we adopt the classical perceptual grouping position and assume only mid-level shape priors. In the relevant work on this topic, such priors can range from simple smoothness to compactness to convexity to symmetry to more elaborate part models, but stop short of object models.

The non-accidental regularity of convexity to group contours into convex parts has been explored by Jacobs [3] and by Estrada and Jepsen [4], to name just two examples. Stahl and Wang [5] explored the non-accidental regularity of symmetry to group contours into symmetric parts, while Lindeberg [6] has explored symmetry to extract symmetric blobs and ridges directly from image data. Although a particular non-accidental shape regularity is exploited by each of these models, they also restrict the image domain. Furthermore, there is little to unify the approaches, since each mid-level shape prior comes with its own computational model.

The early recognition-by-parts paradigm yielded more powerful part models. Pentland [7] partitioned a binary image into 2-D parts corresponding to the projections of a vocabulary of 3-D deformable superquadrics. His method was never applied to contours, since its main focus was more on the problem of part selection (from a large set of part hypotheses) than the grouping of features into parts. Dickinson et al. [8] used part-based aspects (representing the possible views of a vocabulary of volumetric parts) to cover the contours in an image. Pilu and Fisher [9], sought to recover 2-D deformable part models from image contours. Nonetheless, all these approaches were restricted to scenes containing very simple objects, since they assumed a one-to-one correspondence between image and model contours. These systems achieved little, if any, true abstraction and were rarely, if ever, applied to textured objects.

Fitting part models to regions is the dual problem of fitting part models to contours. A method to find instances of a 2-D shape (possibly a part model) in an image was proposed by Liu and Sclaroff [10]. Taking as input a bottom-up image region segmentation, they explore the space of region merges and splits, searching for region groups whose shapes are similar to a 2-D statistical template model. Also starting with a bottom-up region segmentation, Wang et al.'s approach [11] searches for region groups having a particular shape via a stochastic framework that explores the space of region merges and splits. These approaches, however, not only admit a single model shape, but their grouping process is heavily driven by appearance homogeneity. Furthermore, Wang et al.'s method does not attempt shape abstraction, employing a very detailed model of the shape.

Although we know of no approaches dealing with the problem of finding spatiotemporally coherent perceptual groups, this can be considered, in a sense, to be similar to the tracking problem. Tracking approaches often require some type of initialization to indicate the location, in an initial frame, of the region or object of interest that is to be tracked. Moreover, if during the tracking process the tracker's focus of attention drifts away from the objects of interest, some recovery mechanism needs to be in place to recover from such errors. Our method, however,

requires neither an initialization nor a drift-recovery step, since the hypothesis detection process applied at each frame acts as an interest operator, yielding the set of image regions of interest in each frame.

The solution proposed in this paper to the problem of determining multiple sequences (i.e., trajectories) of closed contours, each corresponding to the boundary of a particular object surface across frames, is formulated in graph-theoretical and probabilistic terms, and solved efficiently using the Viterbi algorithm. Quach and Farooq [12] have applied Viterbi to solve the data association problem for single-target tracking in a maximum likelihood fashion, assuming that object motion is a Markov process. More recently, Yan et al. [13] have used Viterbi for single-target tracking of a tennis ball in video. These approaches only admit a single-target, and require both an initialization step and a step to identify the object of interest at the end of the sequence. Our method, however, is not only multi-target, modeling both shape and appearance to disambiguate surface correspondences across frames, but also does not require any type of initialization or recovery mechanism. Moreover, our formulation models second-order relationships between the position, orientation and scale of the surface contours across frames rather than simply modeling first-order smoothness of the tracked feature’s location across frames.

3 Overview of the Approach

The input to our perceptual grouping framework is a video sequence and a vocabulary of shape primitives. First, hypotheses are independently recovered from each frame using the method proposed in [1]. Specifically, we begin by computing a region oversegmentation (Figure 1(b)) of the frame (Figure 1(a)). The resulting region boundaries yield a *region boundary graph* (Figure 1(c)), in which nodes represent region boundary junctions where three or more regions meet, and edges represent the region boundaries between nodes; the region boundary graph is a multigraph, since there may be multiple edges between two nodes. We cast the problem of grouping regions into perceptually coherent shapes as finding simple cycles in the region boundary graph whose shape is “consistent” with one of the model shapes in the input vocabulary (Figure 1(d)); these are called *consistent cycles*. Since the number of simple cycles in a planar graph [14] is exponential, simply enumerating all cycles (e.g., [15]) and comparing their shapes to the model shapes is intractable. Instead, we start from an initial set of single-edge paths and extend these paths (see Section 4.1), called *consistent paths*, as long as their shapes are consistent with a part of *some* model. To determine whether a certain path is consistent (and therefore extendable), the path is approximated at multiple scales with a set of polylines (piecewise linear approximations), and each polyline is classified using a one-class classifier trained on the set of training shapes (Figure 1(e)). When a consistent path is also a simple cycle, it is added to the set of output consistent cycles (Figure 1(f)).

Figure 1(d) shows the input vocabulary used in our experiments: four part classes (superellipses plus sheared, tapered, and bent rectangles, representing the rows) along with a few examples of their many within-class deformations

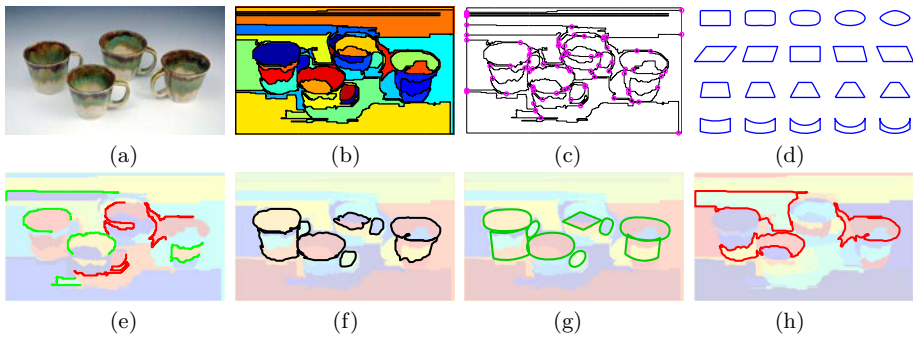


Fig. 1. Problem Formulation: (a) input image; (b) region oversegmentation; (c) region boundary graph; (d) example vocabulary of shape models (used in our experiments); (e) example paths through the region boundary graph that are consistent (green) and inconsistent (red); (f) example detected cycles that are consistent with some model in the vocabulary; (g) abstractions of cycles consistent with some model; (h) example cycles inconsistent with all models

(representing the columns). Each shape model is allowed to anisotropically scale in the x - and y -directions, rotate in the image plane, and vary its deformation parameters (e.g., shearing, tapering, bending).

The algorithm outputs cycles of contours that are consistent with one of the model (training) shapes. However, as mentioned in Section II, the consistent cycle classifier may yield many false positives at reasonable recall rates. Some of the recovered consistent cycles may yield shapes that are qualitatively different from those in the vocabulary, while in other cases the shapes may be consistent but accidental, e.g., a number of the detected consistent cycles might not correspond to actual scene surfaces. By exploiting spatiotemporal consistency of these consistent cycles across a video sequence, we can filter out many of these false positives. That is, we assume that the only cycles that are likely to be caused by the projection of an actual scene surface are those whose shape and internal appearance remain stable or vary smoothly across consecutive frames.

We formulate the problem of finding sequences of consistent cycles with temporally coherent shapes across frames of a video sequence in graph-theoretical and probabilistic terms. We refer to such sequences as *trajectories*. The potential correspondences between consistent cycles detected at different frames are modeled by constructing a graph in which a maximum-weight path corresponds to a trajectory with maximum joint probability of including all and only those consistent cycles in the sequence that correspond to the same scene’s surface boundary. Specifically, nodes in the graph encode pairs of potential matches between consistent cycles in nearby frames, edges connect pairs of nodes that share a common consistent cycle, and edge weights encode the probability of correctness of the cycle matches connected by the edge conditioned on geometric and photometric properties of the cycles involved. We learn this probability distribution from a few hand-labeled training sequences. The top trajectories of temporally coherent consistent cycles are obtained by iteratively

applying the Viterbi algorithm on the graph to find paths with maximum joint probability, and removing from the graph the nodes involved in such paths.

4 Detecting Consistent Cycles

In the following subsections, we review the steps of our algorithm, described in [1], for finding consistent cycles in a single frame, i.e., cycles whose shape is consistent with one of the model shapes. The two main steps of the algorithm are *path initialization* and *path extension*. In Section 5, we introduce the temporal coherence constraint to our grouping framework.

4.1 Path Initialization

The first step in the algorithm generates an initial set of single-edge paths that will be iteratively extended into cycles by repeated executions of the path extension step. This set of edges should be as least redundant as possible, to avoid generating the same cycle more than once (from different edges in the same cycle). Moreover, all possible graph cycles should be realizable by path extensions starting from edges in this set. Such an optimal set corresponds to the *feedback edge set*, which is the smallest set of edges whose deletion results in an acyclic graph. This initial set of single-edge paths are added to the queue of paths to be extended.

4.2 Path Extension

At each algorithm iteration, one of the paths is taken off the queue. If the path is a cycle and it is consistent with at least one of the shapes in the vocabulary of model shapes, the cycle is added to the output list of consistent cycles. If, however, the path is not a cycle, its consistency is also checked. If the path is consistent with a portion of the boundary of at least one shape in the vocabulary, then the path's possible extensions by a single edge are added to the queue. The algorithm continues until the queue is empty, and then outputs the consistent cycles.

Consistency of a cycle or path is checked by first approximating the shape of the cycle or path with a polyline computed at different scales using the Ramer-Douglas-Peucker algorithm [16]. For each resulting polyline, a feature vector is computed, encoding the angles and normalized lengths of the linear segments making up the polyline. As illustrated in Figure 2 (a), a feature vector's length is a function of the number of linear segments comprising the polyline. A consistency decision for a feature vector is made by a one-class classifier that determines if the feature vector is geometrically close to one of the training feature vectors. (Notice that since the feature vectors can have different sizes depending on the lengths of their corresponding polylines¹, there is a classifier for each possible feature vector length.) The scales at which their corresponding polylines

¹ The number K of linear segments comprising the longest polyline approximating a model's contour is determined by the shapes in the vocabulary and the "level of abstraction" (i.e., tolerance, proportional to model size), used to compute the polyline approximations of training model fragments. In our implementation, $K = 13$.

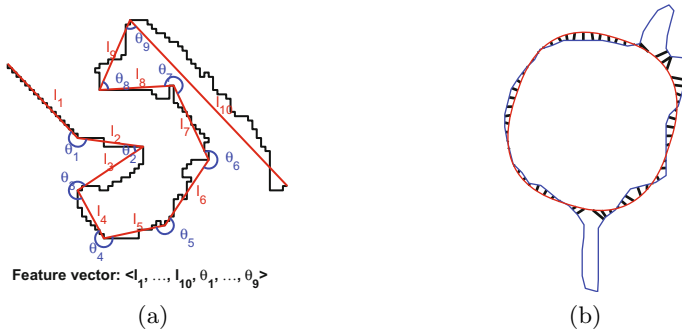


Fig. 2. (a) Feature vector computation for a polyline approximation of a contour; (b) Model-based abstraction (red) of a consistent cycle hypothesis (blue): the black line segments illustrate the distance between equidistantly sampled model points to their closest points along the hypothesis' contour

are consistent are associated with the path. If a path at a particular scale is not consistent, then no extension of that path can be consistent at that scale. Thus, when a path is initialized, it is associated with all scales, and when it is extended, its associated scales can only remain constant or decrease. If there is no scale at which the path is consistent, the path is discarded.

4.3 Training the Classifiers

We trained the classifiers using feature vectors generated from approximately 4 million contour fragments of noisy instances of within-class deformations of each model. Feature vectors are generated from the polyline approximations (computed using a tolerance proportional to model size) of each sampled contour fragment and their dimensionality is reduced via PCA. Classification is performed on the reduced dimensionality vectors. For the model vocabulary employed in our experiments, 99% of the feature vector variance is, in general, captured by the top N PCA components for the case of feature vectors of dimension $2N - 1$, corresponding to polylines with N linear segments. We obtained very fast classification and good accuracy using as classifiers a Nearest Neighbor Data Description approach [17].

5 Temporal Coherence of Consistent Cycles

We formulate the problem of finding temporally coherent consistent cycles in a video sequence in graph-theoretical terms as the search for maximum-weight paths between the source and sink nodes of a particular directed graph $G = (V, E)$. In order to obtain a more robust model of consistent cycle correspondence across frames, we not only model the first derivative of a cycle's pose function (i.e., the cycle's frame-to-frame change in position, scale and orientation), but we also model its second derivative, i.e., the change in the pose transformation function between corresponding consistent cycles. For this reason, instead of modeling the problem via a trellis graph, in which nodes represent

consistent cycles and edges model potential cycle correspondences across frames, we actually define G as the dual of such a graph. Namely, nodes represent potential matches between consistent cycles detected in close spatial and temporal proximity, and there is an edge between each pair of nodes that share a common consistent cycle. Two special nodes, a source and a sink, also exist, which are connected to every other node in G . Edge weights correspond to a log-probability conditional to various attributes of the cycles involved in the edge, such that a maximum-weight path from source to sink corresponds to the trajectory with the highest joint probability of containing the densest sequence of correct consistent cycle matches.

5.1 Retrieving Consistent Cycle Trajectories

The construction of graph G is as follows. The set V contains two special nodes, s and t , called *source* and *sink*, respectively. All other nodes in V correspond to potential matches between consistent cycles detected at different frames and are referred to as *internal*. Formally, if \mathcal{C}_i is the set of all consistent cycles detected at frame i , the set of nodes V is defined as $V = V^{\text{internal}} \cup \{s, t\}$, where $V^{\text{internal}} \subset \bigcup_{i < j} \mathcal{C}_i \times \mathcal{C}_j$. An internal node involving consistent cycles x and y is noted by $\langle x, y \rangle$. There is an edge connecting every pair of nodes that share a common consistent cycle. The direction of these edges, referred to as *internal* edges, is determined by the frame numbers at which the non-common consistent cycles in the pair were detected. Namely, edges leave from the nodes whose non-common consistent cycles are detected at earlier frames. There is also a directed edge from s to every internal node, as well as directed edges from all internal nodes to t . The former edges called *initial*, while the latter are called *final*. Formally, $E = E^{\text{initial}} \cup E^{\text{internal}} \cup E^{\text{final}}$, where $E^{\text{initial}} = \{(s, \langle x, y \rangle) : \langle x, y \rangle \in V\}$, $E^{\text{internal}} = \{(\langle x, y \rangle, \langle y, z \rangle) : \langle x, y \rangle, \langle y, z \rangle \in V \text{ and } n(x) < n(z)\}$, and $E^{\text{final}} = \{(\langle x, y \rangle, t) : \langle x, y \rangle \in V\}$, where $n(x)$ denotes the frame at which cycle x was detected.

A match $\langle x, y \rangle$ is said to be *correct* iff consistent cycles x and y correspond to projected boundaries of the same image surface. The cardinality of V (and thus the total running time of the algorithm) can be kept low by not including in V^{internal} cycle correspondences that are highly unlikely to be correct. This can be done by assuming that a cycle undergoes smooth changes in location, scale, shape, and appearance across frames. Therefore, potential matches can be considered only between cycles whose distance along these dimensions falls within given threshold values proportional to the distance between the frames in which they were detected. Also, consideration can be restricted to matches of cycles detected at frames that are within a specified maximum frame distance W . This maximum frame distance should be chosen such that the likelihood of a consistent cycle being undetected (e.g., due to undersegmentation) for that many consecutive frames is low.

We model the change in appearance between two potentially corresponding cycles by first approximating the shape of one of the cycles by a polygon whose vertices are points sampled at equidistant positions along the cycle. The cycle's

internal appearance is then modeled by computing a homogeneous triangulation of the polygon (e.g., a Delaunay triangulation constraining triangle angles and areas to ensure an approximately uniform sampling of the image region inside the cycle at a fine enough resolution). The triangulation is then mapped onto the other cycle by means of the estimated geometrical transformation between the cycles, and their appearance distance is measured in terms of the absolute difference between sampled image color values at the centroids of corresponding triangles.

From all trajectories of consistent cycles corresponding to some particular scene surface, we are interested in finding the trajectory that is the *densest*, i.e., the one that does not miss any frame where a consistent cycle accounting for the specific surface exists. A correct match $\langle x, y \rangle$ is said to be *consecutive* iff no consistent cycle corresponding to the same surface boundary as x and y was detected in a frame $k : n(x) < k < n(y)$. Let $x \sim y$ represent the relation “ $\langle x, y \rangle$ is a correct and consecutive match”, and let $\neg b(x)$ ($\neg a(x)$) symbolize the predicate “no consistent cycle that correctly matches x was detected before (after) frame $n(x)$.” If $\langle x_i, x_j \rangle$ is a potential match, then T_{ij} represents the geometric transformation between cycles x_i and x_j . The weight $w(\cdot)$ of an edge is a log conditional probability defined depending on the type of edge:

$$w((s, \langle x_1, x_2 \rangle)) = \log(p(\neg b(x_1))p(x_1 \sim x_2 | \theta_{12})) \tag{1}$$

$$w(\langle \langle x_1, x_2 \rangle, \langle x_2, x_3 \rangle \rangle) = \log(p(x_2 \sim x_3 | x_1 \sim x_2, \phi_{123})) \tag{2}$$

$$w(\langle \langle x_1, x_2 \rangle, t \rangle) = \log(p(\neg a(x_2))), \tag{3}$$

where $\theta_{ij} = \langle \mathbf{t}_{ij}, \delta n_{ij}, \delta sh_{ij} \rangle$ and $\phi_{ijk} = \langle \mathbf{t}_{jk}, \delta n_{jk}, \delta sh_{jk}, \delta T_{ijk} \rangle$ are attributes of the consistent cycles involved in the edge. Namely, $\mathbf{t}_{ij} \in \mathbb{R}^2$ is the change in contour position between x_i and x_j , $\delta n_{ij} = |n(x_j) - n(x_i)|$, δsh_{ij} is the shape distance between cycles x_i and x_j , and δT_{ijk} is the difference between the transforms T_{ij} and T_{jk} computed at each consistent cycle correspondence.

With this edge weight specification, a path $(s, \langle x_1, x_2 \rangle, \dots, \langle x_{r-1}, x_r \rangle, t)$ from source to sink achieving maximum weight corresponds to the trajectory of consistent cycles x_1, \dots, x_r maximizing the probability

$$p(\neg b(x_1))p(\neg a(x_r))p(x_1 \sim x_2 | \theta_{12}) \prod_{i=2}^{r-1} p(x_i \sim x_{i+1} | x_{i-1} \sim x_i, \phi_{i-1, i, i+1}). \tag{4}$$

Now, under the following natural assumptions:

1. $f \sim g$, $\neg b(f)$, and $\neg a(g)$ are mutually independent,
2. $x_i \sim x_j$ and $\phi_{k,l,m}$ are independent if $i \neq l$ or $j \neq m$, and
3. $x_i \sim x_j$ and $\theta_{l,m}$ are independent if $i \neq l$ or $j \neq m$,

equation 4 is equivalent to the joint probability

$$p\left(\neg b(x_1), x_1 \sim x_2 \sim \dots \sim x_r, \neg a(x_r) | \theta_{12}, \{\phi_{i-1, i, i+1}\}_{i=2}^{r-1}\right), \tag{5}$$

thus yielding x_1, \dots, x_r as the trajectory of consistent cycles most likely to be the longest and densest trajectory of correct consistent cycle correspondences

in the video sequence. Trajectories of consistent cycles can thus be efficiently generated in decreasing order of probability by iteratively applying the Viterbi algorithm [18] on G to find the maximum-weight path from s to t , and then removing from V all internal nodes belonging to such a path.

Due to undersegmentation errors in the low-level region segmentation of a frame n , which is the input to the consistent cycle detector, it is possible that no consistent cycle is detected in frame n that corresponds to a surface boundary for which consistent cycles have been indeed detected in nearby frames. In these cases, the retrieved trajectories will be missing the frames in which the undersegmentation occurred. A surface’s position and shape can however be interpolated in a missing frame from its known position and shape in nearby trajectory frames. In our approach, we compute an initial guess for the position and shape of the surface boundary in frame n by linearly interpolating the transformation between the corresponding detected consistent cycles in the closest frames around n . This guess is refined by optimizing the normalized cross-correlation between the image data internal to the consistent cycle in a nearby frame where it was detected, and the image data inside a 2-D window around the initial position estimate in frame n . The surface boundary is thus interpolated into frame n , unless the image appearance inside the contour in the estimated position of frame n and the contour appearance in the closest frames differs significantly. In that case, the surface is assumed to be occluded in frame n .

5.2 Probability Density Estimation

In order to compute the edge weights, we need to model the probability distributions involved in Equations 1, 2 and 3. By applying Bayes’ rule, the probability function from Equation 1, $p(x_1 \sim x_2 | \theta_{12})$, can be rewritten as

$$\frac{p(\theta_{12} | x_1 \sim x_2) p(x_1 \sim x_2)}{p(\theta_{12} | x_1 \sim x_2) p(x_1 \sim x_2) + p(\theta_{12} | x_1 \approx x_2) p(x_1 \approx x_2)}, \tag{6}$$

and the probability function $p(x_2 \sim x_3 | x_1 \sim x_2, \phi_{123})$ from Equation 2 as:

$$\frac{p(\phi_{123} | x_2 \sim x_3, x_1 \sim x_2) p(x_2 \sim x_3, x_1 \sim x_2)}{p(\phi_{123} | x_2 \sim x_3, x_1 \sim x_2) p(x_2 \sim x_3, x_1 \sim x_2) + p(\phi_{123} | x_2 \approx x_3, x_1 \sim x_2) p(x_2 \approx x_3, x_1 \sim x_2)}. \tag{7}$$

We can thus estimate these probability distributions from training sequences.

Notice that we can factor $p(\theta_{12} | x_1 \bowtie x_2)$ as

$$p(\mathbf{t}_{12} | \delta n_{12}, \delta sh_{12}, x_1 \bowtie x_2) p(\delta n_{12}, \delta sh_{12} | x_1 \bowtie x_2), \tag{8}$$

where $\bowtie \in \{\sim, \approx\}$. In our experiments, we quantized the space of $(\delta n_{12}, \delta sh_{12})$ values, discretely modeling $p(\delta n_{12}, \delta sh_{12} | x_1 \bowtie x_2)$ via a probability table. And $p(\mathbf{t}_{12} | \delta n_{12}, \delta sh_{12}, x_1 \bowtie x_2)$ (for each quantized value of $(\delta n_{12}, \delta sh_{12})$) was modeled by a multivariate Gaussian, which appeared to be a good approximation to this distribution. Analogously, $p(\phi_{123} | x_2 \bowtie x_3, x_1 \sim x_2)$ can be factored as

$$p(\mathbf{t}_{23}, \delta T_{123} | \delta n_{23}, \delta sh_{23}, x_2 \bowtie x_3, x_1 \sim x_2) p(\delta n_{23}, \delta sh_{23} | x_2 \bowtie x_3, x_1 \sim x_2), \tag{9}$$

and so we modeled $p(\delta n_{23}, \delta sh_{23} | x_2 \bowtie x_3, x_1 \sim x_2)$ by a probability table, and $p(\mathbf{t}_{23}, \delta T_{123} | \delta n_{23}, \delta sh_{23}, x_2 \bowtie x_3, x_1 \sim x_2)$ by a multivariate Gaussian distribution for each quantized value of $(\delta n_{23}, \delta sh_{23})$. The value of $p(x_2 \bowtie x_3, x_1 \sim x_2)$ is computed directly from the training sequences. Finally, we approximated $p(-b(x))$ by $q^{n(x)-1}$ and $p(-a(x))$ by $q^{F-n(x)}$, where F is the total number of frames in the sequence and q is a tight lower bound of $p(x \sim y | n(y) = n(x) + 1)$ computed from the training sequences.

6 Results

We are not aware of any benchmark dataset for evaluating spatiotemporal contour grouping using abstract part models. Therefore, to evaluate our proposed approach, we generated an annotated dataset consisting of 12 video sequences² (a total of 484 frames), containing object exemplars whose 3-D shape can be qualitatively described by cylinders, bent or tapered cubic prisms, and ellipsoids. The visible surface contours of each object’s 3-D shape that are consistent with 2-D models from our vocabulary were hand-labeled.

Figures 3 and 4 illustrate the output of our approach on two selected frames (closer to the beginning and end) of six sequences in the dataset: row (a) shows the input frames; row (b) shows the consistent cycles closest to the ground-truth detected at each static frame (obtained by 1); row (c) shows the temporally coherent detected consistent cycles closest to the ground-truth; and row (d) shows the ground-truth surface contours. Notice that images in rows (c) and (d) also show the boundaries of the region oversegmentation used as input to 1 (computed using the “statistical region merging” approach of Nock and Nielsen 19 with its parameters fixed for all frames from all sequences). The numbers in the top-right corner of each image in rows (b) and (c) correspond to the total number of consistent cycles in each case. The numbers appearing in the centroid of the recovered hypotheses in these rows indicate the rank of the hypothesis among all recovered hypotheses in the frame. In the case of static consistent cycle detection, such ranking is a function of the fitting error between the consistent cycle and the model abstracting the cycle³. In the spatiotemporal case, hypotheses are ranked by the length of the consistent cycle’s temporal flow (i.e., the number of frames in which the cycle is found to be temporally consistent).

These ranking values were obtained after a non-maximum suppression step was applied to eliminate redundant cycle hypotheses in the static and dynamic cases, by discarding all but one of the similar consistent cycles competing for

² Available at <http://www.cs.toronto.edu/~psala/datasets.html>

³ Abstraction of a cycle’s contour by a model in the vocabulary is accomplished via a robust active shape model fitting framework. (See 1 for details.) A hypothesis is ranked based on the average distance from equidistantly sampled points along the abstracting model’s contour to their closest points on the hypothesis’ contour, normalized by the mean distance from the hypothesis’ centroid to its contour. (See Figure 2 (b).) (As in 1, a significant portion of the hypothesis’ contour has to be explained by the model for an abstraction to be considered correct.)

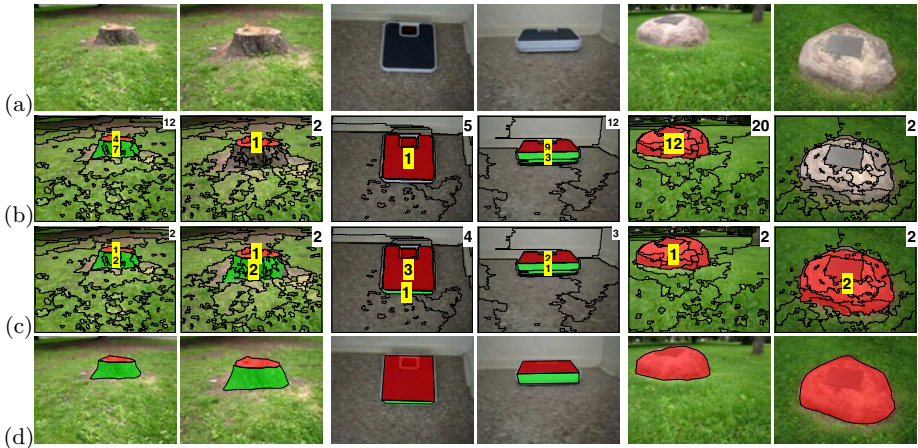


Fig. 3. Part Recovery (see text for discussion)

the same image evidence. (The cycle achieving the smallest shape distance to all other competing cycles was kept.) In the static case, as in [11], detected hypotheses with a high fitting error to their abstraction shapes were also discarded. By comparing the rankings of the recovered hypotheses corresponding to ground-truth parts in the static (row (b)) and dynamic (row (c)) cases, we can see that employing temporal coherence outperforms the static version, as the rankings in row (c) are consistently higher than those in row (b). In some cases, even the rankings of ground-truth parts in row (c) correspond to the top ones. Moreover, the total number of candidate hypotheses in the static case is generally higher than in the dynamic version, demonstrating the superior performance of the dynamic approach to prune false positive hypotheses.

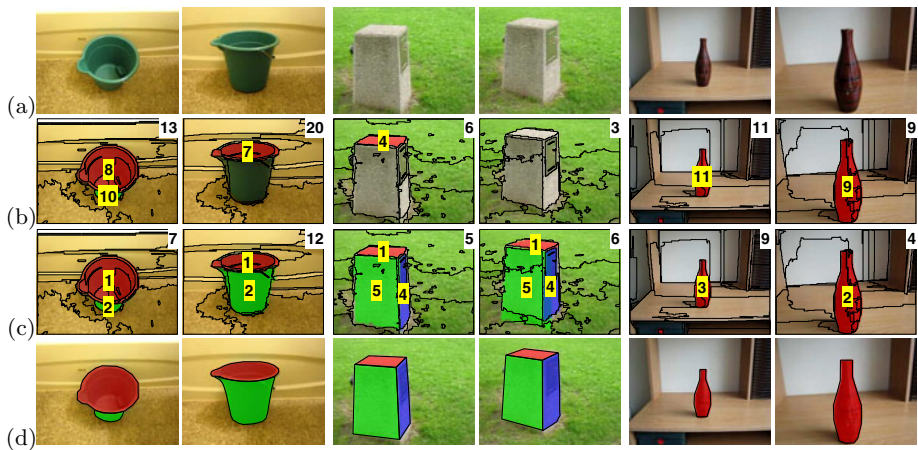


Fig. 4. Part Recovery (cont'd - see text for discussion)

A quantitative evaluation of our spatiotemporal grouping framework is shown in the precision-recall curves of Figure 5, where it is compared to [1] as a baseline. There, it can be seen that both precision and recall increase substantially when temporal coherence is taken into account. The increase in precision can be explained as the result of the pruning ability of our temporal coherence framework on false positive consistent cycles. Since such hypotheses are produced by accidental arrangements of texture or image structure in a single frame, they are unlikely to be temporally stable. Moreover, in the spatiotemporal case, hypotheses are ranked by their persistence, which proves to be a better measure of hypothesis relevance than ranking by the fitting error between a consistent cycle’s contour and its model abstraction contour, as employed in the static case. The improved recall is the result of interpolating hypotheses when gaps of false negatives (mostly due to undersegmentation) have a length not greater than the maximum frame distance W used in the construction of graph G . (In our experiments, $W = 6$.) In terms of running time, the entire process of searching for consistent cycle trajectories in a video sequence takes an average time of less than 5 seconds per frame, in our MATLAB implementation running on a laptop.

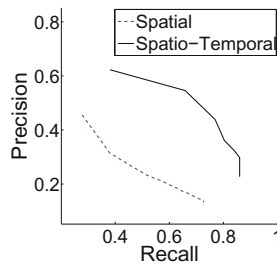


Fig. 5. Quantitative Evaluation: Precision-recall curve (see text for discussion)

7 Conclusions

The semantic gap between real scene contours and the abstract parts that make up categorical shape models can be bridged with the help of a small vocabulary of part models. Yet as the degree of abstraction between image contours and abstract parts increases, so too does the ambiguity of a perceptual group of image contours – if abstraction is viewed as a process of “controlled hallucination”, the more you hallucinate, the greater the possible mappings to different parts. By imposing spatiotemporal constraints on the grouping process, we can significantly reduce such ambiguity, ensuring greater precision of the recovered abstract parts which, in turn, facilitates the indexing and recognition of categorical shape models.

References

1. Sala, P., Dickinson, S.: Contour grouping and abstraction using simple part models. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6315, pp. 603–616. Springer, Heidelberg (2010)

2. Ferrari, V., Fevrier, L., Jurie, F., Schmid, C.: Groups of adjacent contour segments for object detection. *PAMI* 30, 36–51 (2008)
3. Jacobs, D.W.: Robust and efficient detection of salient convex groups. *PAMI* 18, 23–37 (1996)
4. Estrada, F., Jepson, A.: Perceptual grouping for contour extraction. In: *ICPR* (2004)
5. Stahl, J., Wang, S.: Globally optimal grouping for symmetric boundaries. In: *CVPR* (2006)
6. Lindeberg, T.: Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention. *IJCV* 11, 283–318 (1993)
7. Pentland, A.P.: Automatic extraction of deformable part models. *IJCV* 4, 107–126 (1990)
8. Dickinson, S.J., Pentland, A.P., Rosenfeld, A.: 3-d shape recovery using distributed aspect matching. *PAMI* 14, 174–198 (1992)
9. Pilu, M., Fisher, R.: Model-driven grouping and recognition of generic object parts from single images. In: *ISIRS*, Lisbon, Portugal (1996)
10. Liu, L., Sclaroff, S.: Deformable model-guided region split and merge of image regions. *IVC* 22, 343–354 (2004)
11. Wang, J., Gu, E., Betke, M.: Mosaicshape: Stochastic region grouping with shape prior. In: *CVPR* (2005)
12. Quach, T., Farooq, M.: Maximum likelihood track formation with the viterbi algorithm. In: *CDC*, Lake Buena Vista, FL, pp. 271–276 (1994)
13. Yan, F., Christmas, W., Kittler, J.: A maximum a posteriori probability viterbi data association algorithm for ball tracking in sports video. In: *ICPR*, Hong Kong, pp. 279–282 (2006)
14. Buchin, K., Knauer, C., Kriegel, K., Schulz, A., Seidel, R.: On the number of cycles in planar graphs. In: Lin, G. (ed.) *COCOON 2007*. LNCS, vol. 4598, pp. 97–107. Springer, Heidelberg (2007)
15. Tiernan, J.C.: An efficient search algorithm to find the elementary circuits of a graph. *Commun. ACM* 13, 722–726 (1970)
16. Douglas, D., Peucker, T.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *CC* 10, 112–122 (1973)
17. Tax, D., Duin, R.: Data description in subspaces. In: *ICPR*, vol. 2, pp. 672–675 (2000)
18. Forney, G.D.: The viterbi algorithm. *Proceedings of the IEEE* 61, 268–278 (1973)
19. Nock, R., Nielsen, F.: Statistical region merging. *PAMI* 26, 1452–1458 (2004)

Efficient Multi-structure Robust Fitting with Incremental Top- k Lists Comparison

Hoi Sim Wong, Tat-Jun Chin, Jin Yu, and David Suter

School of Computer Science,
The University of Adelaide, South Australia
{hoi.wong,tjchin,jin.yu,david.suter}@adelaide.edu.au

Abstract. Random hypothesis sampling lies at the core of many popular robust fitting techniques such as RANSAC. In this paper, we propose a novel hypothesis sampling scheme based on incremental computation of distances between partial rankings (top- k lists) derived from residual sorting information. Our method simultaneously (1) guides the sampling such that hypotheses corresponding to all true structures can be quickly retrieved and (2) filters the hypotheses such that only a small but very promising subset remain. This permits the usage of simple agglomerative clustering on the surviving hypotheses for accurate model selection. The outcome is a highly efficient multi-structure robust estimation technique. Experiments on synthetic and real data show the superior performance of our approach over previous methods.

1 Introduction

Robust model fitting techniques play an integral role in computer vision since the observations or measurements are frequently contaminated with outliers. Major applications include the estimation of various projective entities from multi-view data [1] which often contain false correspondences. At the core of many robust techniques is random hypothesis generation, i.e., iteratively generate many hypotheses of the geometric model from randomly sampled minimal subsets of the data. The hypotheses are then scored according to a robust criterion (e.g., RANSAC [2]) or clustered (e.g., Mean Shift [3]) to find the most promising model(s). Success rests upon retrieving an adequate number of *all-inlier* minimal subsets which may require a large enough number of sampling steps.

This paper addresses two major issues affecting the current paradigm of robust estimation. The first is that hypothesis generation tends to be time consuming for heavily contaminated data. Previous methods attempted to improve sampling efficiency by guiding the sampling such that the probability of selecting all-inlier minimal subsets is increased. These methods often depend on assumptions or domain knowledge of the data, e.g., inliers have higher keypoint matching scores [4, 5] or are correspondences that respect local geometry patterns [6]. Most methods, however, are not optimized for data with *multiple instances* (or *structures* [7]) of the geometric model. This is because they sample based on

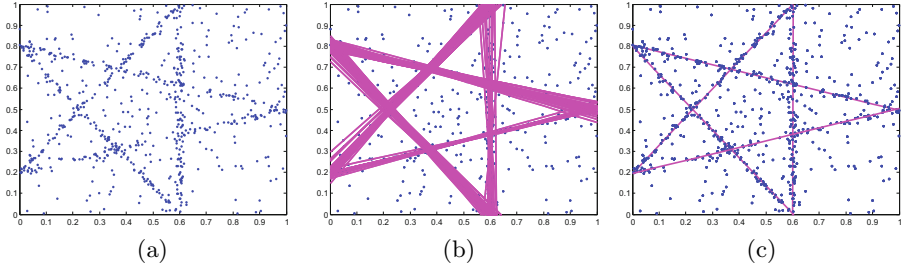


Fig. 1. (a) Input data with 5 structures (lines) with 100 points per structure and 250 gross outliers. The inlier scale is 0.01. (b) 500 hypotheses are generated with the proposed *multi-structure* guided sampling scheme and *simultaneous* hypothesis filtering, producing 146 good hypotheses as shown in the figure. (c) Simple agglomerative clustering of the remaining 146 hypotheses gives the final fitting result.

estimated inlier probabilities alone while ignoring the fact that only inliers from the *same* structure should be included in the same minimal subset. Such methods may inefficiently generate a large number of samples before obtaining an all-inlier minimal subset for each genuine structure in the data.

The second crucial issue is the lack of a principled approach to fit the multiple structures in the data. Many previous works [8,9] simply apply RANSAC sequentially, i.e., fit one structure, remove corresponding inliers, then repeat. This is risky because inaccuracies in the initial fits will be amplified in the subsequent fits [10]. Moreover, finding a stopping criterion for sequential fitting that accurately reflects the true number of structures is non-trivial. Methods based on clustering [11] or mode detection [3,12] given the generated hypotheses are not affected by the dangers of sequential fitting. However, if there are insufficient hypotheses corresponding to the true structures, the genuine clusters will easily be overwhelmed by the irrelevant hypotheses. Consequently, these methods often miss the true structures or find spurious structures.

The inability to retrieve “good” hypotheses at sufficiently large quantities represents the fundamental obstacle to the satisfactory performance of previous methods. To address this limitation, we propose a novel hypothesis sampling scheme based on incremental computation of distances between *partial rankings* or *top- k lists* [13] derived from residual sorting information. Our approach enhances hypothesis generation in two ways: (1) The computed distances guide the sampling such that inliers from a *single* coherent structure are more likely to be simultaneously selected. This dramatically improves the chances of hitting all-inlier minimal subsets for *each* structure in the data. (2) The qualities of the generated hypotheses are evaluated based on the computed distances. This permits an on-the-fly filtering scheme to reject “bad” hypotheses. The outcome is a set of only the most promising hypotheses which facilitate a simple agglomerative clustering step to fit all the genuine structures in the data. Fig. 1 summarizes the proposed approach.

The rest of the paper is organized as follows: Sec. 2 describes how to derive data similarities from residual sorting information by comparing top- k lists.

Sec. 3 describes our guided sampling scheme with simultaneous hypothesis filtering and incremental computations of distances between top- k lists. Sec. 4 describes how multi-structure fitting can be done by a simple agglomerative clustering on the promising hypotheses returned by our method. Sec. 5 presents results on synthetic and real data which validate our approach. Finally, we draw conclusions in Sec. 6.

2 Data Similarity by Comparing Top- k Lists

A key ingredient of our guided sampling scheme is a data similarity measure. This section describes how to derive such a measure from residual sorting information.

2.1 Top- k Lists from Residual Sorting Information

We measure the similarity between two input data based on the idea that if they are inliers from the same structure, then their preferences to the hypotheses as measured by residuals will be similar. Such preferences can be effectively captured by lists of ranked residuals.

Let $X = \{x_i\}_{i=1}^N$ be a set of N input data and $\theta = \{\theta_j\}_{j=1}^M$ a set of M hypotheses, where each hypothesis θ_j is fitted from a minimal subset of p points (e.g., $p=2$ for line fitting). For each datum x_i , we compute its absolute residual $r_i = \{r_1^{(i)}, r_2^{(i)}, \dots, r_M^{(i)}\}$ as measured to M hypotheses. We sort the elements in r_i to obtain the list of sorted residual $\tilde{r}_i = \{r_{\lambda_1^{(i)}}^{(i)}, \dots, r_{\lambda_M^{(i)}}^{(i)}\}$ such that $r_{\lambda_1^{(i)}}^{(i)} \leq \dots \leq r_{\lambda_M^{(i)}}^{(i)}$. The top- k list of data x_i is defined as the first k elements in the permutation $\{\lambda_1^{(i)}, \dots, \lambda_M^{(i)}\}$, i.e.,

$$\tau_i = \{\lambda_1^{(i)}, \dots, \lambda_k^{(i)}\}. \tag{1}$$

The top- k list τ_i essentially gives the top- k hypotheses preferred by x_i , i.e., x_i is more likely to be an inlier to the hypotheses which have higher rank.

2.2 The Spearman Footrule Distance

Given the top- k lists, we measure their similarity using the Spearman Footrule (SF) distance [13]. Let τ be a top- k list and D_τ a set of elements contained in τ . Denote the position of the element $m \in D_\tau$ in τ by $\tau(m)$. The SF distance between two top- k lists τ_i and τ_j is defined as

$$F^{(\ell)}(\tau_i, \tau_j) = \sum_{m \in D_{\tau_i} \cup D_{\tau_j}} |\tau_i'(m) - \tau_j'(m)|, \tag{2}$$

where $\ell > 0$ is the so-called location parameter (often set to $k+1$), $\tau_i'(m) = \tau_i(m)$ if $m \in D_{\tau_i}$; otherwise $\tau_i'(m) = \ell$, and τ_j' is similarly obtained from τ_j .

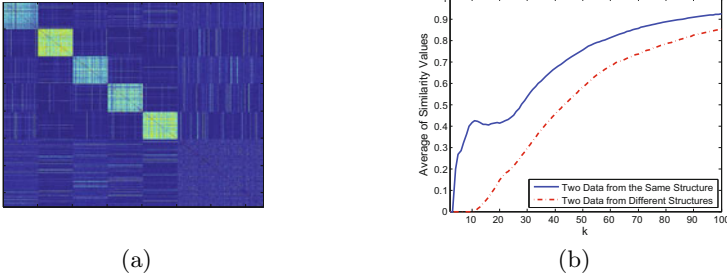


Fig. 2. (a) Similarity matrix K for data shown in Fig. 1a (data is arranged according to structure membership for representation only) (b) The average of similarity values between two data from the same structure and two data from different structures under various k

2.3 Measuring Similarity between Data

To measure the similarity between two data, we use the SF distance (Eq. 2) between their corresponding top- k lists. The similarity value between two data x_i and x_j is defined as

$$d(\tau_i, \tau_j) = 1 - \frac{1}{k \times \ell} F^{(\ell)}(\tau_i, \tau_j). \quad (3)$$

Note that we normalize $F^{(\ell)}(\tau_i, \tau_j)$ such that $d(\tau_i, \tau_j)$ is between 0 (dissimilar) and 1 (identical). By comparing the top- k lists between all data, we obtain a $N \times N$ similarity matrix K with

$$K(i, j) = d(\tau_i, \tau_j), \quad (4)$$

where $K(i, j)$ denotes the element at its i -th row and j -th column. Fig. 2a shows an example of K , which is generated from the input data shown in Fig. 1a. The evident block structures in K correspond to the 5 lines in Fig. 1a. As shown in Fig. 2b, across a wide range of k , the similarity value between two data from the same structure (solid) is higher than that from different structures (dotted).

3 Guided Sampling with Hypothesis Filtering

This section describes our guided sampling scheme which involves a simultaneous hypothesis filtering scheme. We also provide an efficient incremental update for computing the sampling weights.

3.1 Guided Sampling

We use the similarity matrix K (Eq. 4) to sample data in a guided fashion. Let $Q = \{s_u\}_{u=1}^p$ be the indices of data in a minimal subset of size p , where s_u are indexed by the order in which they are sampled. The first element s_1 in Q is

randomly selected from X . To sample the next element s_2 , we use $K(s_1, \cdot)$ as the weight to guide the sampling, i.e., the similarity values of all input data with respect to s_1 . We set $K(s_1, s_1)$ to 0 to avoid sampling the same data again.

Suppose data s_1, \dots, s_u have been selected, then the next datum s_{u+1} is chosen conditionally on the selected data. Its sampling weight is defined as

$$K'(s_1, \cdot) \cdot K'(s_2, \cdot) \cdot \dots \cdot K'(s_u, \cdot), \tag{5}$$

where \cdot is the element-wise multiplication and $K'(s_u, \cdot)$ is just $K(s_u, \cdot)$ with $K(s_u, s_u) = 0$. Eq. 5 means that in order to have higher probabilities of being sampled, a datum need to be similar (measured by Eq. 3) to all the data that have been selected into the minimal subset.

3.2 Incremental Top- k Lists Comparison

Our sampling method computes an update to the similarity matrix K (Eq. 4) once a block (of size b) of new hypotheses are generated. This involves comparing top- k lists of ranked residuals. The computation of top- k lists can be done efficiently via merge sort. However, comparing top- k lists between all data, i.e., constructing K , can be computationally expensive. Here we provide efficient incremental updates for K that can substantially accelerate the computation.

As proved in [13], the SF distance (Eq. 2) can be equivalently computed as

$$F^{(\ell)}(\tau_i, \tau_j) = 2(k - |Z|)\ell + \sum_{m \in Z} |\tau_i(m) - \tau_j(m)| - \sum_{m \in S} \tau_i(m) - \sum_{m \in T} \tau_j(m), \tag{6}$$

where $Z = D_{\tau_i} \cap D_{\tau_j}$, $S = D_{\tau_i} \setminus D_{\tau_j}$ and $T = D_{\tau_j} \setminus D_{\tau_i}$. In fact, S is simply the elements in D_{τ_i} but not in Z , i.e., $S = D_{\tau_i} \setminus Z$, similarly for T . Hence, we have

$$\sum_{m \in S} \tau_i(m) = \sum_{m=1}^k m - \sum_{m \in Z} \tau_i(m) = \frac{1}{2}k(k+1) - \sum_{m \in Z} \tau_i(m), \tag{7}$$

similarly for $\sum_{m \in T} \tau_j(m)$. By setting $\ell = k + 1$ and using Eq. 7, we can rewrite Eq. 6 to be in terms of Z only,

$$F^{(k+1)}(\tau_i, \tau_j) = (k+1)(k-2|Z|) + \sum_{m \in Z} (|\tau_i(m) - \tau_j(m)| + \tau_i(m) + \tau_j(m)). \tag{8}$$

Let A and B be two $N \times N$ symmetric matrices with zero on diagonal, and set the elements at the i -th row and the j -th column of A and B to

$$A(i, j) = |Z| \text{ and } B(i, j) = \sum_{m \in Z} (|\tau_i(m) - \tau_j(m)| + \tau_i(m) + \tau_j(m)). \tag{9}$$

From Equations 3, 4, and 8, the similarity matrix K can be constructed by

$$K = 1 - \frac{1}{k}(kI_N - 2A) - \frac{1}{k(k+1)}B, \tag{10}$$

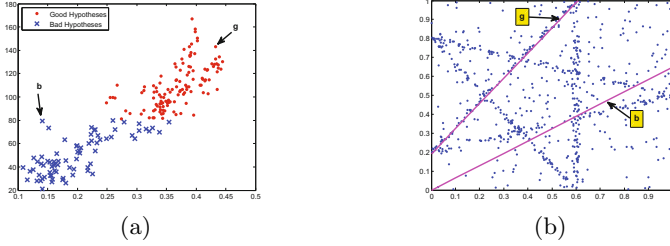


Fig. 3. (a) Feature space where x axis is $f_m^{(1)}$ and y axis is $f_m^{(2)}$ (Best view in color). (b) An example of “good” (denoted “g”) and “bad” (denoted “b”) hypotheses.

where I_N is an $N \times N$ identity matrix. Observe from Eq. 9 that the matrices A and B can be efficiently updated by keeping track of the elements that move into or out of Z . This information is readily available from the merge sort. Once A and B are updated, K can be updated via Eq. 10.

3.3 Simultaneous Hypothesis Filtering

During sampling, we want to simultaneously filter hypotheses such that only a small but very promising subset remains. Let $R_m = \{R_m^{(1)}, \dots, R_m^{(N)}\}$ be the absolute residual of N input data as measured to a hypothesis m . For this hypothesis, we construct a feature vector

$$f_m = \left[f_m^{(1)}, f_m^{(2)} \right] = \left[\frac{\sum_{(i,j) \in E} K(i,j)}{|E|}, \frac{|\Omega_m|}{\sum_{\{i|x_i \in \Omega_m\}} R_m^{(i)}} \right], \tag{11}$$

where $E = \{(i,j) | i \neq j \text{ and } x_i, x_j \in \Omega_m\}$ with $\Omega_m = \{x_i \in X | m \in D_{\tau_i}\}$, and K is the similarity matrix computed by Eq. 10. The set Ω_m contains all data that include the hypothesis m in their top- k lists. If the hypothesis m is “good”, then Ω_m should contain many inliers from a structure. Hence, $f_m^{(1)}$, the average of similarity values between all data in Ω_m should be high. Moreover, the average residual of data in Ω_m should be low, i.e., high $f_m^{(2)}$. Therefore, we want to find a set of hypotheses which have high value in both $f_m^{(1)}$ and $f_m^{(2)}$. To this end, we apply k-means on the feature vectors (Eq. 11) to separate “good” and “bad” hypotheses. As illustrated in Fig. 3(a), the cluster whose center has larger norm (dots) contains good hypotheses. We incrementally maintain a set of “good” hypotheses as the guided sampling proceeds.

4 Multi-structure Fitting

By leveraging the simultaneous hypothesis filtering, a set of “good” hypotheses is immediately available once the sampling is done. The minimal subsets of these “good” hypotheses should mainly contain inliers from different structures.

Hence, we can perform the final fitting by first, clustering the minimal subsets of “good” hypotheses, and then fitting geometric models to each cluster of data.

We use the agglomerative clustering (See [14] for a detailed description) to cluster the minimal subsets. First, we need to define a distance measure between two minimal subsets. From Fig. 2a, we can see that if two data x_i and x_j are from the same structure, the rows $K(i, :)$ and $K(j, :)$ of the similarity matrix K must have higher values on the same dimension, implying that the sum of the element-wise multiplication of these two rows must have higher value. Based on this observation, we assign to each minimal subset Q_u a $1 \times N$ feature vector

$$\alpha_u = K(s_1^u, :) \cdot K(s_2^u, :) \cdot \dots \cdot K(s_p^u, :), \quad (12)$$

where \cdot is the element-wise multiplication and s_1^u, \dots, s_p^u are indices of data in Q_u . The distance between two minimal subsets Q_u and Q_v is given by

$$d(Q_u, Q_v) = \frac{1}{\|\alpha_u \cdot \alpha_v\|_1}, \quad (13)$$

where $\|\cdot\|_1$ denotes the L_1 norm. Using this distance measure, the clustering is then performed through the standard agglomerative clustering mechanism.

5 Experiments

We test the proposed method (ITKSF) on several synthetic and real datasets. To evaluate the efficiency of the proposed guided sampling scheme, we compare our method against 6 sampling techniques: Uniform random sampling in RANSAC (Random) [2], proximity sampling (Proximity) [9, 11], LO-RANSAC [15], Guided-MLESAC [4] and PROSAC [5].

In all experiments, the scale parameter of Proximity (σ^2 as in Equation 1 in [11]) is set to twice the squared average nearest neighbor distance. For LO-RANSAC, the inlier threshold is set to the average residual of inliers as measured to their corresponding structures. For PROSAC, T_N is set to 5×10^4 . For our method, we fix $b = 10$ and $k = \lceil 0.1 \times t \rceil$ throughout, b being the block size (cf Sec. 3.2) and t the number of hypotheses generated so far. All experiments are run on a machine with 2.53GHz Intel Core 2 Duo processor and 4GB RAM.

5.1 Multiple 2D Line and Circle Fitting

We test the performance of various sampling methods on multiple 2D line and circle fitting under various numbers of gross outliers. Fig. 4 (Left) shows the test data. The inliers scale is set to 0.01, and the number of inliers per structure is 50 for lines and 80 for circles. We simulate the quality score required by PROSAC and Guided-MLESAC by probabilistically assigning higher scores to inliers than gross outliers. Each method is given 50 random runs, each for 2 CPU seconds.

As can be seen in Fig. 4 (Second column), in all cases the average percentage of all-inlier samples found by ITKSF within the given time budget is significantly

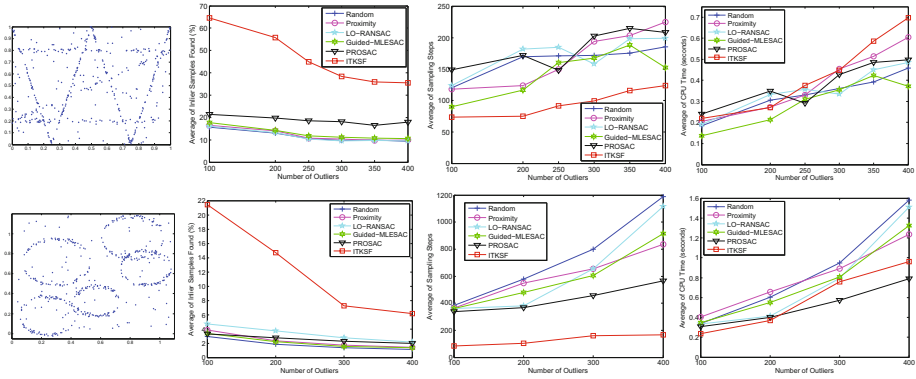


Fig. 4. Performance of various sampling methods on 2D lines and circles data under various numbers of gross outliers. First column: input data. Second column: the average percentage of all-inlier samples found within 2 CPU seconds. Third column: the average sampling steps (respectively CPU time, last column) needed to hit at least one all-inlier minimal subset for each structure.

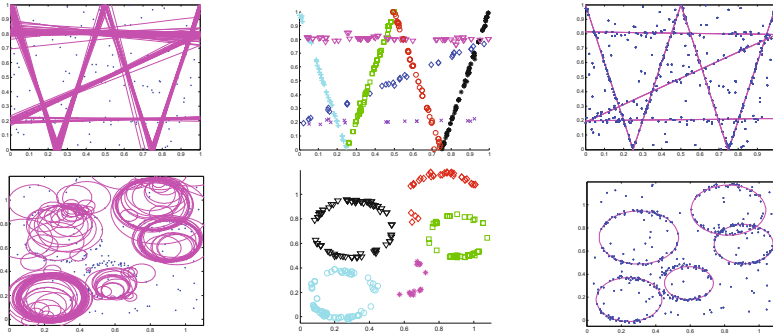


Fig. 5. Left: “good” hypotheses returned by ITKSF after sampling. Center: clusters found by clustering the minimal subsets of “good” hypotheses. Right: final fitting result.

higher than its competing methods. It also consistently requires less sampling steps than other methods to hit at least one all-inlier minimal subset for each structure (Third column). In terms of CPU time, ITKSF exhibits comparable performance to other methods (Fig. 4. Last column). Note that ITKSF simultaneously picks “good” hypotheses during sampling (Sec. 3.3). The CPU time spent on this operation is counted toward the reported CPU time for ITKSF.

Fig. 5 (Left) shows the “good” hypotheses returned by ITKSF after sampling. It is evident that they are concentrated on the genuine structures present in data. Fig. 5 (Center) shows that the minimal subsets of these promising hypotheses are correctly clustered by the agglomerative clustering (Sec. 4). Model fitting on each individual cluster of data then leads to the final fitting results shown in Fig. 5 (Right).

5.2 Homography Estimation

Our second set of experiments involves estimating multiple planar homographies on real images data.¹ For each image pair, keypoint correspondences (including false correspondences) and their matching scores are generated by SIFT matching² [16]. The image data with marked keypoint correspondences can be found in Table 1, the false correspondences are marked as yellow crosses. We use 4 correspondences to estimate a homography using Direct Linear Transformation [1]. Each method is given 50 random runs, each for 15 CPU seconds.

Table 1 summarizes our experimental results. We can see that ITKSF outperforms other methods across all performance measures in almost all cases; only on the College II data, it is slightly slower than PROSAC in terms of CPU time. Noticeably, our method finds much more all-inlier samples within the given time budget (Structures) than its competing methods. The average percentage of all-inlier samples (IS) found by ITKSF is up to an order of magnitude higher than other methods. Moreover, it is one of the only two methods that succeed in finding at least an all-inlier sample for each structure in all 50 runs.

Fig. 6 shows that the minimal subsets of the hypotheses provided by our guided sampling procedure are indeed inliers, and they are correctly clustered according to their membership to a planar structure. Given these clusters, we can obtain multiple homographies by model fitting on each cluster of data.

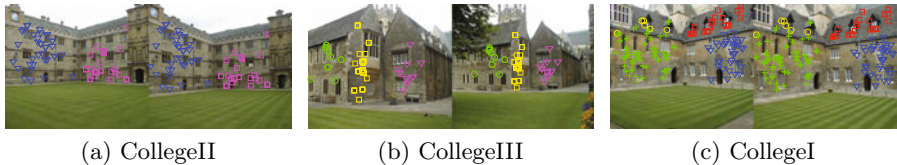


Fig. 6. Clusters found by clustering the minimal subsets of the “good” hypotheses returned by our method (Best view in color)

5.3 Fundamental Matrix Estimation

We now evaluate the performance of various sampling methods for the task of fundamental matrix estimation on the Hopkins data.³ The image data with marked keypoint correspondences (obtained from SIFT matching) are shown in Table 2. We use the standard 7-point algorithm [1] to estimate the fundamental matrix.⁴ Each method is given 50 random runs, each for 30 CPU seconds.

In Table 2, we can see that in the case of single fundamental matrix estimation (on the Truck data) PROSAC is the most effective sampling method in terms of

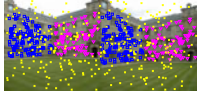
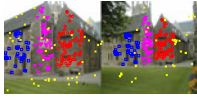

¹ <http://www.robots.ox.ac.uk/~vgg/data>

² Code from <http://www.vlfeat.org/~vedaldi/code/sift.html>

³ <http://www.vision.jhu.edu/downloads/data/hopkins155/>

⁴ <http://www.robots.ox.ac.uk/~vgg/hzbook/code/>

Table 1. Performance of various sampling methods in multiple homographies estimation. We record the average CPU time (Time) (respectively sampling steps (Steps)) required and the number of random runs a method fails (Fail) to hit at least one all-inlier minimal subset for each structure. We also report the average percentage of all-inlier samples found within the given time budget (IS). The average number of all-inlier samples found for each structure (Structures) is separately listed in square bracket. The reported result is taken over successful runs only with the best result boldfaced.

Data	Sampling Method	Time (seconds)	Steps	Fail	Structures	IS (%)
 (a) CollegeII	Random	1.03	298	0	[109,25]	1.84
	Proximity	0.58	163	0	[137,33]	2.48
	LO-RANSAC	0.46	146	0	[162,37]	2.56
	Guided-MLESAC	0.5	179	0	[131,31]	2.6
	PROSAC	0.23	86	0	[354,86]	7.46
	ITKSF	0.25	45	0	[337,162]	33.74
 (b) CollegeIII	Random	2.27	945	0	[6,20,125]	2.24
	Proximity	0.69	253	0	[18,21,151]	2.99
	LO-RANSAC	1.66	641	0	[6,22,148]	2.66
	Guided-MLESAC	1.74	685	0	[7,20,247]	4.21
	PROSAC	1.71	677	0	[6,12,189]	3.19
	ITKSF	0.31	75	0	[228,98,305]	28.81
 (c) CollegeI	Random	2.82	2080	15	[5,12,3,2]	0.37
	Proximity	8.72	953	0	[20,57,13,6]	1.69
	LO-RANSAC	2.82	1750	3	[7,21,5,3]	0.61
	Guided-MLESAC	4.91	1855	3	[4,17,8,7]	0.60
	PROSAC	5.17	1819	2	[6,18,6,4]	0.57
	ITKSF	1.36	198	0	[74,93,30,8]	14.4

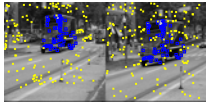

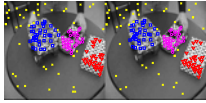
CPU time required to find an all-inlier minimal subset for the single structure, while ITKSF performs best in terms of the average number of all-inlier samples found within the given time budget. Overall, all sampling enhancement methods are effective on this simple single structure recovery task.

We now move to the more challenging case where more than one structure is present. Table 2 shows that previous methods fail disastrously on the Cars⁵ and the Toy Cars data, which contain 2 and 3 structures, respectively. All previous methods fail to hit an all-inlier sample for each structure in 24%-100% of the given 50 runs, while ITKSF succeeds in every run. The average CPU time required by ITKSF to find at least one all-inlier sample for each structure is about 80% less than the best-performing competing method. Within the given time budget, the overall number of all-inlier samples found by ITKSF is again substantially larger than other methods. For instance, on the Toy Cars data, the average percentage of all-inlier samples found by ITKSF is at least an order of magnitude higher than others.

Fig. 7 shows the clusters found by clustering the minimal subsets of the “good” hypotheses returned by ITKSF. It can be seen that each cluster is formed of inliers to one structure present in data. The fundamental matrix for each structure can therefore be effectively obtained by model fitting on each cluster of data.

⁵ The Cars data originally contains 3 structures. We use two in our experiment in order to create different levels of difficulties in the three datasets used in our experiments.

Table 2. Performance of various sampling methods in fundamental matrix estimation. The same notations as used in Table 1 are used here.

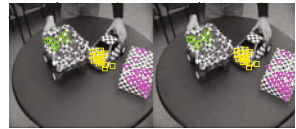
Data	Sampling Method	Time (seconds)	Steps	Fail	Structures	IS (%)
 (a) Truck	Random	5.94	1156	0	[5]	0.08
	Proximity	1.73	318	0	[16]	0.26
	LO-RANSAC	0.35	78	0	[10]	0.15
	Guided-MLESAC	0.23	62	0	[165]	3.23
	PROSAC	0.008	1	0	[58]	0.93
	ITKSF	0.2	27	0	[471]	16.24
 (b) Cars	Random	×	×	50	[0,0]	0
	Proximity	13.2	3997	41	[67,1]	0.69
	LO-RANSAC	13.63	4227	49	[25,1]	0.25
	Guided-MLESAC	6.94	3796	47	[334,1]	3.08
	PROSAC	×	×	50	[0,0]	0
	ITKSF	2.19	450	0	[813,14]	16.77
 (c) Toy Cars	Random	9.92	3015	49	[1,1,1]	0.02
	Proximity	15.09	4448	12	[5,2,3]	0.07
	LO-RANSAC	20.63	6092	48	[2,2,3]	0.04
	Guided-MLESAC	17.85	6662	36	[3,2,1]	0.04
	PROSAC	6.16	2436	36	[2,2,1]	0.03
	ITKSF	1.93	305	0	[244,28,11]	8.11



(a) Truck



(b) Cars



(c) Toy Cars

Fig. 7. Clusters found by clustering the minimal subsets of the “good” hypotheses returned by our method (Best view in color)

6 Conclusions

We propose a novel guided sampling scheme based on the distances between top- k lists that are derived from residual sorting information. In contrast to many existing sampling enhancement techniques, our method does not rely on any domain-specific knowledge, and is capable of handling multiple structures. Moreover, while performing sampling, our method simultaneously filters the hypotheses such that only a small but very promising subset remains. This permits the use of simple agglomerative clustering on the surviving hypotheses for accurate model selection. Experiments on synthetic and real data show the superior performance of our approach over previous methods.

References

1. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004)

2. Fischler, M.A., Bolles, R.C.: RANSAC: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM* 24, 381–395 (1981)
3. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence* 24, 603–619 (2002)
4. Tordoff, B.J., Murray, D.W.: Guided-MLESAC: Faster image transform estimation by using matching priors. *TPAMI* 27, 1523–1535 (2005)
5. Chum, O., Matas, J.: Matching with PROSAC- progressive sample consensus. In: *CVPR* (2005)
6. Sattler, T., Leibe, B., Kobbelt, L.: SCRAMSAC: Improving RANSAC's efficiency with a spatial consistency filter. In: *ICCV* (2009)
7. Stewart, C.V.: Robust parameter estimation in Computer Vision. *SIAM Review* 41, 513–537 (1999)
8. Vincent, E., Laganiere, R.: Detecting planar homographies in an image pair. In: *Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis, ISPA 2001*, pp. 182–187 (2001)
9. Kanazawa, Y., Kawakami, H.: Detection of planar regions with uncalibrated stereo using distributions of feature points. In: *BMVC* (2004)
10. Zuliani, M., Kenney, C., Manjunath, B.: The multiransac algorithm and its application to detect planar homographies. In: *IEEE International Conference on Image Processing, ICIP 2005*, vol. 3 (2005)
11. Toldo, R., Fusiello, A.: Robust multiple structures estimation with j-linkage. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part I. LNCS*, vol. 5302, pp. 537–547. Springer, Heidelberg (2008)
12. Xu, L., Oja, E., Kultanen, P.: A new curve detection method: randomized Hough transform (RHT). *Pattern Recognition Letters* 11, 331–338 (1990)
13. Fagin, R., Kumar, R., Sivakumar, D.: Comparing Top shapek Lists. In: *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, vol. 36. SIAM, Philadelphia (2003)
14. Hastie, T., Tibshirani, R., Friedman, J.: *The Elements of Statistical Learning*. Springer, Heidelberg (2009)
15. Chum, O., Matas, J., Kittler, J.: Locally optimized RANSAC. In: Michaelis, B., Krell, G. (eds.) *DAGM 2003. LNCS*, vol. 2781, pp. 236–243. Springer, Heidelberg (2003)
16. Lowe, D.: Distinctive image features from scale-invariant keypoints. *IJCV* 60, 91–110 (2004)

Flexible Online Calibration for a Mobile Projector-Camera System

Daisuke Abe, Takayuki Okatani, and Koichiro Deguchi

Graduate School of Information Sciences, Tohoku University

Abstract. This paper presents a method for calibrating a projector camera system consisting of a mobile projector, a stationary camera, and a planar screen. The method assumes the projector to be partially calibrated and the camera to be uncalibrated, and does not require any fiducials or natural markers on the screen. For the system of geometrically compensating images projected on the screen from a hand-held projector so that the images will always be displayed at a fixed position of the screen in a fixed shape, the method makes the projected images geometrically rectified; that is, it makes them have the correct rectangular shape of the correct aspect ratio. The method automatically performs this calibration online without requiring any effort on the user's part; all the user has to do is project a video from the hand-held projector. Furthermore, when the system makes discontinuous temporal changes such as the case where the camera and/or the screen is suddenly relocated, it automatically recovers the calibrated state that was once lost. To realize these properties, we adopt the sequential LS method and extend it to be able to deal with temporal changes of the system. We show several experimental results obtained by a real system.

1 Introduction

The systems combining projectors with cameras, called projector-camera systems, have been widely studied [1–12]. In these systems, images projected by projectors are captured by cameras and then some information extracted from the images is returned to the projectors; such feedback paths between cameras and projectors are used to realize desired properties. There are many applications; examples include geometric as well as photometric correction of projected images and displaying high-resolution images using multiple projectors.

Recently, along with the downsizing of image projecting devices, small size projectors, or *mobile projectors*, have become available in the market; there are also PDAs, mobile phones, and digital cameras having an embedded projection device. Such small projectors can make full use of unique characteristics of projectors that are not shared by other image display devices, and using these projectors for a projector camera system, further new applications are expected to be realized.

One of the central issues in many such applications is the calibration of projector-camera systems. In this paper we deal with a problem of calibrating a type of systems in which images are projected onto a planar screen and a



Fig. 1. From left to right: Overview of the projector-camera system. Example of unrectified images. Results obtained by our method; correctly rectified video images are being projected independently of the projector pose. The calibration is automatically performed while a video is projected from the hand-held projector.

stationary camera captures their images. Its main application is real-time compensation of the position and shape of the images on a screen that are projected from a hand-held projector; see Fig. 1. The calibration we consider here is to make the images being compensated on the screen have a geometrically rectified shape; that is, it makes them have a correct rectangular shape of the correct image aspect ratio.

Our goal is to make it possible to perform such calibration without requiring any effort on the user's part. Specifically, i) we assume no fiducials or no natural markers on the screen, and assume the projector to be partially calibrated and the camera to be fully uncalibrated. ii) The calibration should automatically be performed online; all the user has to do is hold the projector by his/her hand and project a video. (The projector needs to be more or less moved for the calibration.) No offline procedure is necessary. iii) Discontinuous temporal changes of the system can be dealt with. When the camera or the screen is relocated, the system should automatically recover the calibrated state in which the projected images are rectified.

We propose a method that satisfies the above requirements (i)-(iii). To satisfy (i), we apply to a mobile video projector the results of the studies on the system that uses multiple stationary projectors to generate a seamless single image [5, 6]. However, the solution is obtained only through nonlinear optimization, where a large number of unknowns need to be determined simultaneously. Therefore, it is not easy to achieve an accurate calibration in a stable manner. To resolve this difficulty and satisfy requirement (ii) of automatic/online calibration, we propose to use the sequential least squares (LS) minimization. Online calibration necessitates using natural images for the calibration, which results in low-quality observations such as limited precision in feature point extraction. Although this can possibly be overcome by using the large number of images, it means increase in computational cost. The use of sequential LS method helps resolve this dilemma, owing to its nature that it improves the accuracy of estimates by processing time-series observations one by one. To satisfy requirement (iii), we extend the sequential LS method to be able to deal with discontinuous system changes by providing it with an adaptive nature.

We have developed a real system and examined the performance of the proposed method. It can perform the sequential LS computation at 5fps or faster,

while a video is being projected from a hand-held projector at about 15fps. We show several experimental results that demonstrate the effectiveness of our approach.

2 Related Work

There exist several studies on the online calibration/recalibration of projector-camera systems. Cotting et al. [7], Zhou et al. [8], and Johnson et al. [9] developed different calibration methods mainly for multi-projector displays. In their studies, they assumed that a projector and a few cameras, which form a functional unit, are mostly stationary and move occasionally. Yang and Welch [10] showed how the shape of a display surface can be estimated online from projector-camera image correspondences, where projector and camera are assumed to be stationary and fully calibrated. Zollman et al. [11] developed a method for correcting the distortion of images projected onto an arbitrary-shaped surface by a system consisting of a stationary projector and a continuously moving camera. Their method performs view-dependent distortion correction.

All the earlier works (except [5, 6, 12]), including the above, either assume projectors/cameras to be internally calibrated or adopt the basic self-calibration technique of cameras for a general (non-planar) 3D scene. However, this approach cannot be applied to our setting wherein the projector and the camera are both uncalibrated and the display surface is planar, since this setting induces degeneracy of self-calibration.

To cope with this difficulty with planar scenes, a specialized calibration method needs to be used. This method was first presented for camera calibration in [13], and it is applied to projector-camera systems in [5, 6, 12]. Our contribution is as follows: 1) the application of this approach to an image-display/human-interface system of a mobile projector, 2) the adoption of sequential LS optimization and its extension to enable the online calibration/recalibration, and 3) the experimental confirmation of the feasibility of the approach.

3 Problem Formulation

3.1 Geometry of the Projector-Camera System

We start with revisiting [12] to formulate the problem to be solved to realize the above calibration method.

As mentioned above, we consider a system in which there are a planar screen, a stationary camera, and a moving projector. We denote each pose of the moving projector by $p = 1, \dots$. We use a 3D coordinate system for each of the projector, the camera, and the screen, as shown in Fig. 2. For the projector and the camera, their 3D coordinates are defined in the usual manner. For the screen, its coordinates are defined such that the xy -plane lies in the screen. Additionally, two image coordinates are defined for the image planes of the projector and the camera.

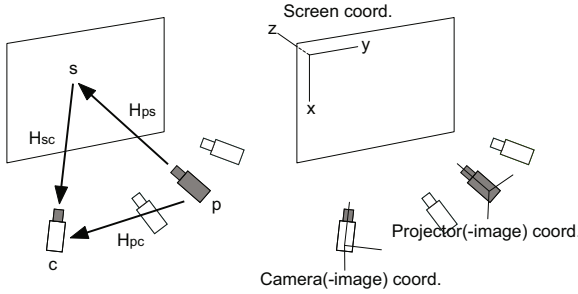


Fig. 2. Left: Three two-dimensional projective transformations between pairs of the screen, the image planes of the camera and of the projector. Right: The three coordinate systems used here.

There are planar projective transformations between a pair of the three planes: the screen and the image plane of the projector and that of the camera. Consider a point \mathbf{u} in the projector image and let \mathbf{w} be its projection on the screen and also \mathbf{v} be its corresponding point on the camera image. These three points are mutually transformed by the following three two-dimensional projective transformations, as shown in Fig. 2:

$$\mathbf{w} \propto H_{ps} \mathbf{u}, \tag{1a}$$

$$\mathbf{v} \propto H_{sc} \mathbf{w}, \tag{1b}$$

$$\mathbf{u} \propto H_{pc} \mathbf{v}, \tag{1c}$$

where \propto represents equality up to scale. The following relation holds for the three projective transformations:

$$H_{ps} \propto H_{sc}^{-1} H_{pc}. \tag{2}$$

The projective transformation H_{ps} is factored into a product of three 3×3 matrices as follows:

$$H_{ps} \propto T_p R_p K_p^{-1}, \tag{3}$$

where T_p encodes the screen coordinates $[x_p, y_p, z_p]^T$ of the position of projector p as follows:

$$T_p = \begin{bmatrix} z_p & 0 & x_p \\ 0 & z_p & y_p \\ 0 & 0 & 1 \end{bmatrix}. \tag{4}$$

Here, R_p represents the orientation of the projector and K_p is the internal matrix of the projector.

3.2 Fiducial-Less Calibration

Calibrating our projector-camera system reduces to determining H_{sc} . Our objective of rectifying projected images can be achieved if H_{ps} is known for each p . The observations that we can use for the calibration are the pairs of a projector

input image and the image (projected on the screen and) captured by the camera. We call the former a projector image and the latter a camera image. Among the above three projective transformations, \mathbf{H}_{pc} can be computed from a pair of the projector image at pose p and the corresponding camera image if four or more point correspondences ($\mathbf{u} \leftrightarrow \mathbf{v}$) are established. Since we are assuming the camera and the screen to be both stationary, \mathbf{H}_{sc} is constant. Considering the relation $\mathbf{H}_{ps} \propto \mathbf{H}_{sc}^{-1} \mathbf{H}_{pc}$, it is found that once \mathbf{H}_{sc} is determined, the desired \mathbf{H}_{ps} can always be computed from \mathbf{H}_{pc} obtained as above.

If there are fiducials on the screen, then \mathbf{H}_{sc} can be directly estimated from them, for example, by using four or more point correspondences between the screen and the camera image. However, requiring these fiducials narrows areas of applications. We can calculate \mathbf{H}_{sc} without such fiducials on the screen [12], which is summarized as follows.

If \mathbf{K}_p is known, \mathbf{H}_{sc} can be calculated in a closed form manner up to inherent ambiguity; \mathbf{H}_{sc} has originally eight degrees of freedom, and only four of them can be determined. The indeterminate four degrees of freedom correspond to a similarity transformation (i.e., the translation, rotation, and scaling) of the projected images.

If all the elements of \mathbf{K}_p are unknown, \mathbf{H}_{sc} cannot be determined. If the projector is *partially calibrated*, more specifically, when only a single element of \mathbf{K}_p is unknown and others are all known, \mathbf{H}_{sc} can be determined. A practically important case is that the focal length of the projector is unknown and varies for each pose. In [5], this is dealt with for the system of multiple stationary projectors. In this partially calibrated case, a closed-form algorithm has not been found for calculating \mathbf{H}_{sc} , and thus the only solution is to perform nonlinear optimization, or the method of bundle adjustment, as is described in the next section. (It is noteworthy that if we further assume the camera to be partially calibrated (i.e., only its focal length are unknown), this reduces the degrees of freedom of \mathbf{H}_{sc} by one).

3.3 Nonlinear Least Squares Optimization

Let \mathbf{u}_{pi} be the i -th feature point of the input image to projector p . Also let \mathbf{v}_{pi} be its corresponding point in the camera image. Since we know the projector images and thus the true value of each \mathbf{u}_{pi} is known, we minimize the sum of reprojection errors of \mathbf{v}_{pi} over all points $i = 1, \dots, n_p$ and all projector poses $p = 1, \dots, m$:

$$E_{1:m}(\mathbf{x}) = \sum_{p=1}^m E_p(\mathbf{x}), \quad (5)$$

where \mathbf{x} is the vector containing the unknown parameters to be estimated; E_p is the sum over all the feature points of projector p :

$$E_p(\mathbf{x}) = \sum_{i=1}^{n_p} \left(\tilde{\mathbf{v}}_{pi} - \tilde{\hat{\mathbf{v}}}_{pi} \right)^2, \quad (6)$$

where the operator \sim represents making a inhomogeneous vector; $\hat{\mathbf{v}}_{pi}$ is the estimate of a measured point \mathbf{v}_{pi} and is written by

$$\hat{\mathbf{v}}_{pi} \propto \mathbf{H}_{sc} \mathbf{H}_{ps} \mathbf{u}_{pi} = \mathbf{H}_{sc} \mathbf{T}_p \mathbf{R}_p \mathbf{K}_p^{-1} \mathbf{u}_{pi}. \quad (7)$$

In \mathbf{x} , we store appropriate representations of \mathbf{H}_{sc} , \mathbf{T}_p , \mathbf{R}_p , and \mathbf{K}_p . The value of \mathbf{x} minimizing $E_{1:m}(\mathbf{x})$ is the solution.

To constrain the above ambiguity in \mathbf{H}_{sc} , we parametrize it as follows. The ambiguity reflects the freedom of defining a 2D coordinate system on the screen, and thus we define it indirectly through a camera image. Choosing two points in the camera image, we denote their coordinates by $[\alpha_1, \beta_1]$ and $[\alpha_2, \beta_2]$. Then, we assume that their corresponding points on the screen have the coordinates $[0, 0]$ and $[1, 0]$, respectively. The two chosen point correspondences constrain \mathbf{H}_{sc} as $\mathbf{H}_{sc}[0, 0, 1]^\top \propto [\alpha_1, \beta_1, 1]^\top$ and $\mathbf{H}_{sc}[1, 0, 1]^\top \propto [\alpha_2, \beta_2, 1]^\top$; \mathbf{H}_{sc} satisfying these two constraints can be parametrized with four parameters x_1 , x_2 , x_3 , and x_4 as

$$\mathbf{H}_{sc} = \begin{bmatrix} \alpha_2(x_1 + 1) - \alpha_1 & x_2 & \alpha_1 \\ \beta_2(x_1 + 1) - \beta_1 & x_3 & \beta_1 \\ x_1 & x_4 & 1 \end{bmatrix}. \quad (8)$$

We perform the above minimization using the Levenberg-Marquardt algorithm. For the sake of later discussions, we summarize the algorithm here. Starting with an initial value \mathbf{x} , the optimal solution is sought for by updating \mathbf{x} in an iterative manner as $\mathbf{x}' = \mathbf{x} + \delta\mathbf{x}$, where $\delta\mathbf{x}$ is the solution to a linear equation $(\mathbf{A} + \lambda\mathbf{I})\delta\mathbf{x} = \mathbf{a}$, where \mathbf{A} and \mathbf{a} are the approximate Hessian and the negative gradient of $E_{1:m}$, respectively; they are given by $\mathbf{A} = (1/2)\mathbf{f}'(\mathbf{x})^\top \mathbf{f}'(\mathbf{x})$ and $\mathbf{a} = -\mathbf{f}'(\mathbf{x})^\top \mathbf{f}$, where \mathbf{f} is defined such that $E_{1:m} = (1/2)\mathbf{f}^\top \mathbf{f}$, and \mathbf{f}' is its derivative wrt. \mathbf{x} .

4 Sequential Least Squares Optimization

As described earlier, our goal is to automatically calibrate the system in an online manner while an arbitrary video is being projected on the screen. To do this, we adopt the sequential least squares (LS) method here.

In the situation where new observations arrive one by one as time elapses, the sequential LS method updates the estimates of objective parameters whenever a new observation arrives, and iterates this data-acquisition/parameter-updating process to improve the parameter estimates. Since it can maintain the number of parameters constant at the expense of some loss of accuracy, and the computational cost is kept small at each updating, the method is suitable to be used in an online manner.

The basic idea of the method is to approximate a part of the cost function by a quadratic function. Assuming we currently have m observations, the sum of the cost given by $E_{1:m}(\mathbf{x}) = \sum_p E_p(\mathbf{x})$ is split into two parts as $E_{1:m} = E_{1:a} + E_{a+1:m}$, and the first part is approximated up to the second order. Assuming

that the first part is a function not of the entire parameter \mathbf{x} but of its part \mathbf{x}_1 as $E_{1:a} = E_{1:a}(\mathbf{x}_1)$, we represent its second order approximation $\hat{E}_{1:a}$ as

$$\hat{E}_{1:a}(\mathbf{x}_1) = \frac{1}{2}(\mathbf{x}_1 - \mathbf{x}_1^*)\mathbf{A}_{11}^*(\mathbf{x}_1 - \mathbf{x}_1^*) + \text{const.}, \quad (9)$$

where \mathbf{x}_1^* and \mathbf{A}^* is the minimum and the Hessian of the cost, respectively. Evaluating the remaining part $E_{a+1:m}(\mathbf{x})$ of the cost as it is (i.e. a nonlinear function), $\hat{E}_{1:a} + E_{a+1:m}(\approx E_{1:m})$ is minimized. When the second nonlinear part is small in size, the minimization can be carried out much faster than the minimization of the original cost.

After the minimization, a part of the second cost $E_{a+1:m}$ is cut out and adjoined to the first cost; it is then approximated up to the second order. Assume, for example, a single term E_{a+1} is chosen for the approximation. Defining other parts of \mathbf{x} than \mathbf{x}_1 as $\mathbf{x}^\top = [\mathbf{x}_1^\top, \mathbf{x}_2^\top, \mathbf{x}_3^\top]$, we assume $E_{a+1} = E_{a+1}(\mathbf{x}_1, \mathbf{x}_2)$ and $E_{a+2:m} = E_{a+2:m}(\mathbf{x}_1, \mathbf{x}_3)$. Once $E_{a+1} = E_{a+1}(\mathbf{x}_1, \mathbf{x}_2)$ is approximated by a quadratic function, \mathbf{x}_2 is no longer necessary to explicitly compute and can be eliminated forever from the system. This is realized by updating \mathbf{x}_1^* and \mathbf{A}_{11}^* as follows: the minimizer \mathbf{x}_1 to $\hat{E}_{1:a}(\mathbf{x}_1) + E_{a+1}(\mathbf{x}_1, \mathbf{x}_2)$ gives the new \mathbf{x}_1^* and \mathbf{A}_{11}^* is updated as

$$\mathbf{A}_{11}^* \leftarrow \mathbf{A}_{11}^* + \mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21}, \quad (10)$$

where \mathbf{A}_{jk} ($j, k = 1, 2$) is a block matrix of the Hessian of E_{a+1} with respect to \mathbf{x}_1 and \mathbf{x}_2 . The updated \mathbf{x}_1^* and \mathbf{A}_{11}^* are propagated to the future. When another new observation (E_{m+1}) arrives, the above process is repeated from the beginning.

We apply the sequential LS method to our problem as follows. For the parameters \mathbf{x}_1 to be maintained forever, we select \mathbf{H}_{sc} , the focal length of the projector, and the radial lens distortions of the projector and the camera. The lens distortions are modeled by polynomial functions having two coefficients. Thus, the size of \mathbf{x}_1 is $4 + 1 + 2 + 2 = 9$. The rest of the parameters in the system, the poses of the projector, are encoded in a usual manner.

In the above summary, we divide the observations into $[1 : a]$ and $[a + 1 : m]$. When the length of the nonlinear part $[a + 1 : m]$ is kept constant, say w , a is chosen as $a = m - w$ and is to increase as m goes. We will refer to the nonlinear part of size w as a *window*. Note that in this case, the projector pose parameters for the latest w poses are maintained in the system and those for earlier poses are eliminated. If $w = 1$, the latest pose is eliminated as soon as the solution is updated; updating \mathbf{x}^* is easy in that case, since the minimizer to the total cost gives \mathbf{x}^* . We will discuss how to choose the size w of the window later.

We compared the sequential method thus obtained with the batch method in terms of computational time and estimation accuracy. The results are omitted here due to lack of space, and are instead presented in a supplemental material with this paper submitted.

5 Adaptation to Discontinuous System Change

We have assumed so far that the screen is fixed and the camera is stationary. Relaxing this assumption to some extent, we consider here the case where their relation makes discontinuous changes while the images are being projected. Examples are the case where the user relocate the camera or the case where the user uses a hand-held cardboard for the screen.

The sequential LS method, which can reduce the computational cost of the batch LS method as described above, is not supposed to be able to deal with such cases. To deal with temporal changes in the system, a special mechanism is necessary. The bottom line for such a mechanism is that even if the calibration is insufficient (i.e., the rectification is inaccurate), the projector needs to be able to keep projecting images in a stable manner although they may have a small distortion. Then, as the user moves the projector, the projected images are gradually rectified.

5.1 Imposing an Upper Bound on the Information Matrix

Based on the theory of maximum likelihood estimation, the matrix \mathbf{A}^* (we will write \mathbf{A}_{11}^* as \mathbf{A}^* from now on) updated according to Eq.(10) is regarded as an (approximated) estimate of the inverse of the variance-covariance of the estimate $\hat{\mathbf{x}}_1$ of \mathbf{x}_1 , i.e., $\text{Var}(\hat{\mathbf{x}}_1)^{-1}$. As a series of observations are processed, the accuracy of $\hat{\mathbf{x}}_1$ increases, which corresponds to that the eigenvalues of $\mathbf{A}^*(= \text{Var}(\hat{\mathbf{x}}_1)^{-1})$ tend to have large numbers.

This mechanism that the accuracy of estimates increases monotonically as time elapses is favorable if the system is time-invariant and highly accurate estimation is necessary. However, it is not fit for our purpose. After a long sequence of observations has been processed, \mathbf{A}^* should become large, meaning that new observations will have a relatively small effect on the estimation. Then, if the system makes a sudden change, the latest observations having information about the new system will not be effectively used. Therefore, it is necessary to make it possible to put more weight on latest observations as compared with earlier ones, or in other words, to forget information from earlier observations.

For this purpose, we propose to impose an upper bound on \mathbf{A}^* . Specifically, when propagating the updated \mathbf{A}^* to the next time step, we modify

$$\mathbf{A}^* \preceq \tilde{\mathbf{A}}^*, \quad (11)$$

where $\mathbf{M} \succeq \mathbf{0}$ indicates \mathbf{M} is positive semi-definite and $\tilde{\mathbf{A}}^*$ is a constant matrix. The procedure for modifying \mathbf{A}^* so that the above constraint will be met is as follows. We first diagonalize the given \mathbf{A}^* as $\mathbf{A}^* = \mathbf{U}\mathbf{D}\mathbf{U}^\top$, where \mathbf{U} is an orthogonal matrix storing eigenvectors as its column vectors and \mathbf{D} is a diagonal matrix storing eigenvalues. Letting $\tilde{\mathbf{D}}$ be the diagonal matrix obtained by replacing every negative element of \mathbf{D} with 0, we reset \mathbf{A}^* as $\mathbf{A}^* \leftarrow \tilde{\mathbf{A}}^* - \mathbf{U}\tilde{\mathbf{D}}\mathbf{U}^\top$.

It is reasonable to incorporate some criterion on desired calibration accuracy to calculate the upper bound $\tilde{\mathbf{A}}^*$. We use an inequality regarding rectification

Table 1. Relation between the window size and averaged processing time per each frame. Measured values for our real system.

Window size	$w = 1$	2	3	4	5
Time (sec.)	0.087	0.19	0.26	0.32	0.40

errors for this purpose; the measure of the errors is defined by combining two terms of rectification errors, averaged angle errors and an error of image aspect ratio, which will be explained later. The resulting $\tilde{\mathbf{A}}^*$ has to depend on the unknown parameter \mathbf{x}_1 as $\tilde{\mathbf{A}}^* = \tilde{\mathbf{A}}^*(\mathbf{x}_1)$, and we plug-in the current estimate $\hat{\mathbf{x}}_1$ to it.

Using this method, a new observation always has a certain weight on the final estimation no matter how long the sequence of earlier observations is. Note that a naive method of multiplying a constant α ($0 < \alpha < 1$) to \mathbf{A}^* as $\mathbf{A}^* \leftarrow \alpha \mathbf{A}^*$ when propagating it to the future will not work. In order for this method to work effectively, it is required that the projector continuously moves and its poses distribute in a somewhat uniform manner. Otherwise, the cost function will degenerate and the solution obtained by the minimization will be unstable; the worst case is that the resulting images are severely distorted.

5.2 Adaptive Control of Window Size

We have confirmed through experiments that the sequential method is by no means inferior to the batch method in terms of accuracy even if the minimum window size $w = 1$ is chosen: see one of the supplemental materials. However, this is not considered to be the case with time-variant systems, because of the following reasons. At the time instant when the system makes discontinuous changes, the cost given by the new observations from then on will have the new minimum at a different point from the old minimum given by the observations before then. At first, the total cost combining the new and old observations will have the minimum at a point more or less near by the old minimum. Since the second-order approximation of the cost is guaranteed to be accurate only in a small neighborhood of the minimum, it is clearly not good to immediately approximate the cost of the new observations; the valuable new observations will be spoiled due to the large approximation errors.

To avoid this, it is necessary to delay making the approximation; this is made possible by enlarging the window size ($w > 1$). As long as estimation accuracy is concerned, it will always be good to use as large a window size as possible, but we need also to consider computational time. Table 1 shows the relation between the window size and processing time that is measured for our real system described in Sec. 6. Since shorter processing time means being able to acquire more observations per unit time, this table reconfirms that it is desirable to set $w = 1$, as long as computational time is concerned.

To resolve this dilemma, we propose to change the window size w adaptively to the observations. When the camera or the screen is relocated in a short length of time, it can be detected online by examining the strength of the reprojection

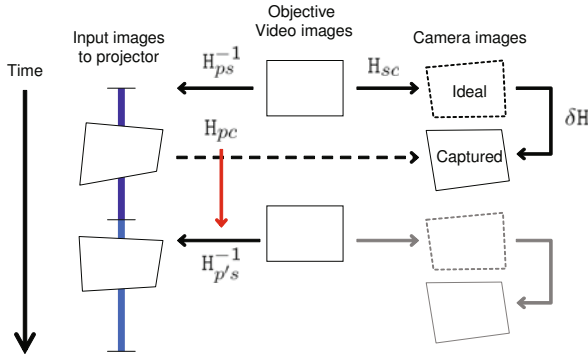


Fig. 3. Flowchart of the projector image compensation

errors for the latest observation, specifically, by thresholding its averaged reprojection error per point. We use this thresholding of reprojection errors as a trigger to control the window size. Let e be the reprojection error per point for the latest image. The algorithm is as follows:

Algorithm 1. Window size control

if $e > T_e$ and $w < w_{\max}$ **then**
 $w \leftarrow w + 1$. (No pose is eliminated.)
else if $e \leq T_e$ and $w > w_{\min}(= 1)$ **then**
 The oldest two poses are eliminated and $w \leftarrow w - 1$.
else
 The oldest one pose is eliminated and w is unchanged.
end if

One might think that it is easier to reset the estimation whenever a system change is detected; the estimation can indeed be reset by discarding \mathbf{A}^* before the change and generating \mathbf{A}^* only from new observations after the change. However, we cannot adopt this approach due to the same reason as above; the solution will be unstable until a sufficient number and variety of projector poses are accumulated. On the other hand, the above approach is expected to balance stability and response speed to system changes.

6 Implementation Details

We implemented a real system to examine the feasibility and usability of the proposed method. Algorithmically, the system consists of a) the compensation of projector images to cancel out projector motion and b) the online calibration based on the point correspondence obtained in the process of (a).

Fig. 3 shows the flowchart of the projector image compensation. The projector image at the next time step is generated from the next frame of the objective

video by warping it with $\mathbf{H}_{p's}^{-1}$. $\mathbf{H}_{p's}^{-1}$ is calculated from the latest projector-camera relation \mathbf{H}_{pc} and the current estimate of the screen-camera relation, \mathbf{H}_{sc} , as $\mathbf{H}_{ps}^{-1} \propto \mathbf{H}_{pc}^{-1} \mathbf{H}_{sc}$. Since directly obtaining \mathbf{H}_{pc} in a limited time is difficult due to large geometric difference between the two images, we employed the method of [4]; we predict the camera image ('Ideal' in Fig. 3) of the projected image and track a small motion $\delta\mathbf{H}$ from it to the real camera image ('Captured'). We then use the other path from the projector image to the camera image to compute \mathbf{H}_{pc} as $\mathbf{H}_{pc} \propto \delta\mathbf{H} \mathbf{H}_{sc} \mathbf{H}_{ps}$.

The prediction of the camera image is performed geometrically as well as photometrically. For this, we estimate the combined response curve from the projector to the camera when starting the image compensation. For the image tracking, we use a GPU implementation [14] of an improved variant of the KLT tracker. To start the image tracking in the beginning, the projector image needs to be identified in the camera image, for which we use a GPU implementation [15] of a SIFT-based image matcher. The GPU is also used for image warping.

The sequential LS method for the calibration is invoked at every three frames of the projector image compensation. The updated \mathbf{H}_{sc} is reflected in the image compensation at the next nearest time step. The calibration requires not the homography but raw point correspondences between the projector and the camera. Therefore, for the point correspondences obtained in the above image tracking of $\delta\mathbf{H}$, we transfer the point coordinates in the predicted camera image to the projector image by $\mathbf{H}_{ps}^{-1} \mathbf{H}_{sc}^{-1}$.

The hardware we use are an Intel Xeon(3MHz) PC with a NVidia Quadro FX 5600 graphics board, a Toshiba TDP-FF1 LED/DLP projector, and a Grasshopper camera of Point Grey Research Inc. The projector is connected to the graphics board via an analog VGA cable and projects images of 800×600 pixels at refresh rate 85Hz. The camera captures images of 640×480 pixels at frame rate 120Hz. We do not synchronize the projector and the camera, and have to put a wait of about 35ms in between the input of an image to the projector and the capture of the associated image by the camera. As a result, the image compensation is carried out about 15fps.

7 Experimental Results

We carried out several experiments using the real system. In the experiments, starting from a situation where images are correctly rectified, the camera is relocated to another pose while a video is being projected on the screen. We observe how image rectification is recovered with time.

We evaluate the accuracy of image rectification using two quantities. One is a measure of angle errors of four corners of a projected image; the RMS value of their deviations from 90 degrees is used. The other is the aspect ratio of a projected image; the ratio of the distances between the midpoints of two opposed sides of the image quadrangle, is used. To calculate these quantities, it is necessary to have the shape of a projected image on the screen. For this purpose, using a planar board having lattice pattern for the screen, we use the image

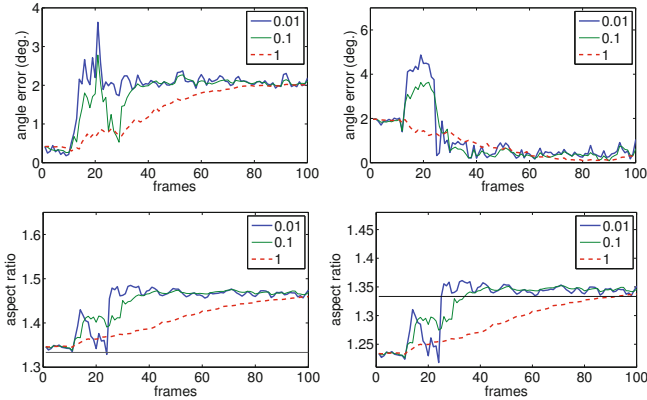


Fig. 4. Angle errors (upper row) and aspect ratios (lower row) of the image rectification. The camera is relocated at the 10th frame in the sequence. The quantities evaluated by using the camera before the relocation are plotted on the left and those by using the camera after the relocation are shown on the right. The horizontal line ($y = 1.333$) on the lower plots indicate the correct aspect ratio.

of the pattern taken by the same camera used for the calibration. Specifically, calculating H_{sc} directly from the image and assuming it to be correct, we transfer the prediction of a projected image on the camera image to the screen and measure its shape.

Fig. 4 shows how image rectification is recovered after a certain relocation of the camera. A video is used for projection for which 100 - 300 points are extracted in each frame. In the sequence, the camera is relocated at about the 10th frame. The upper row shows angle errors and the lower row shows aspect ratios. The left and right columns show the results evaluated using the camera before and after the relocation, respectively. Fig. 5 shows the images of the lattice on the screen taken by the cameras before and after the relocation.

In Fig. 4, three different results are simultaneously plotted for different upper bounds \tilde{A}^* of A^* ; the details are explained later. For all three, it is observed in the left column plots that the angular error is small and the aspect ratio is close to its correct value before the 10th frame, whereas they tend to have large errors after the 10th frame. The reversal is true in the right column plots, where starting from large errors, they gradually become accurate after the 10th frame. Thus, it is seen that image rectification, once lost due to the camera relocation, is recovered by our method. The correctness of the rectification is also confirmed on Fig. 5, where the image shapes of projected images on the screen are drawn.

The three plots in Fig. 4 indicates the results obtained by varying the upper bound of A^* as $\tilde{A}^* = \gamma \tilde{A}_0^*$ with $\gamma = 0.01, 0.1$, and 1.0 , where \tilde{A}_0^* is a constant and determined by the aforementioned method. It is seen from these plots that when γ is small and the upper bound \tilde{A}^* is small, the corner angles as well as the aspect ratio tend to jitter, whereas image rectification is quickly recovered;

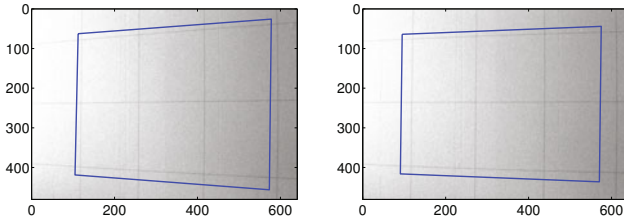


Fig. 5. Overlay of the shapes of projected images (specifically, their predicted camera images) on two camera images before (left) and after (right) the camera relocation. The screen surface has a lattice pattern for validation purpose, using which the accuracy of image rectification can be visually confirmed.

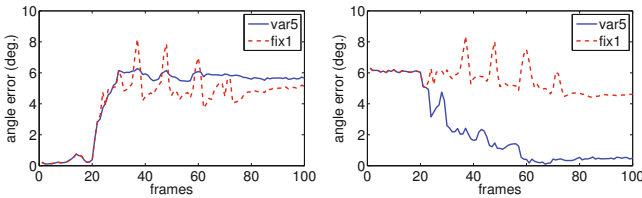


Fig. 6. Angle errors evaluated by the camera before (left) and after (left) its relocation. 'var5' indicates the errors when the window size w is adaptively controlled with $w_{max} = 5$. 'fix1' indicates the errors when $w = 1$.

the reversal is true in the case when γ is small. Thus, considering the balance between stability and response speed to system changes, the user can choose γ .

Fig. 6.8 shows the result for a case where the camera undergoes a larger relocation. The camera is relocated at the 20th frame; the camera images before and after the relocation are shown in Fig. 8. The left and right plots in Fig. 6 show angle errors evaluated by the camera before and after the relocation, respectively. Two results are plotted; one is obtained when the window size is controlled between $w = 1$ and $w_{max} = 5$ as described in Sec. 5.2 (marked as 'var5') and the other is obtained with a fixed window size $w = 1$ (marked as 'fix1'). It is observed that the errors become small about 40 frames later than the camera relocation when the window size is controlled; when it is fixed as $w = 1$, the errors decrease only gradually and have very large values even at the 100th frame. Fig. 7 shows time-series variation of the window size and averaged reprojection error per point. We set the threshold to 1.0 (indicated by a horizontal line in the plot), which determines the trigger level for controlling the window size. These demonstrate the effectiveness of the control of the window size. Fig. 8 shows how the shapes of projected images appear in the camera images before (left) and after (right) of the camera relocation. In the post-relocation camera image on the right, the shape obtained when the window size is controlled is shown in a solid line and the shape obtained for the fixed window size (at the 100th frame) is shown in a dotted line.

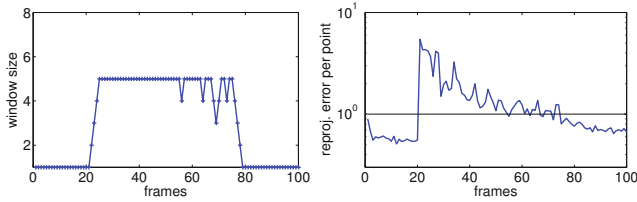


Fig. 7. Left: Time-series variation of the window size w for the sequence of Fig. 6. Right: That of the reprojection error per point. The trigger level used for the window size control is set to 1.0.

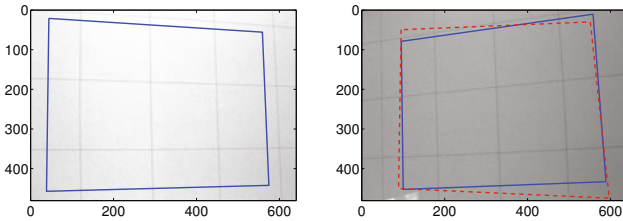


Fig. 8. Shapes of projected images before (left) and after (right) the camera relocation for the sequence of Fig. 6. The dotted line in the right image represents the shape of a projected image generated using H_{sc} obtained for the fixed size window $w = 1$.

8 Summary

We have shown a method for calibrating a projector-camera system consisting of a mobile projector, a stationary camera and a planar screen. When it is used for the system of compensating the position and shape of images projected on the screen from a hand-held projector, the proposed method can make projected images on the screen have the rectangular shape of the correct aspect ratio. It assumes the focal length of the projector and all the internal parameters of the camera to be unknown and does not need fiducials on the screen. The method automatically performs the calibration online without requiring any effort on the user's side. To simultaneously realize high calibration accuracy and small computational cost, we adopt the sequential LS method, and further extend it to be able to deal with discontinuous changes of the system. We have confirmed the effectiveness of our approach through several experiments.

References

1. Raskar, R., Beardsley, P.: A self correcting projector. In: Proc. CVPR, pp. 626–631 (2001)
2. Raskar, R., van Baar, J., Beardsley, P., Willwacher, T., Rao, S., Forlines, C.: ilamps: Geometrically aware and self-configuring projectors. In: Proc. ACM SIGGRAPH (2003)
3. Rehg, J.M., Flagg, M., Cham, T.J., Sukthankar, R., Sukthankar, G.: Projected light displays using visual feedback. In: Proc. International Conference on Control, Automation, Robotics and Vision (2002)

4. Johnson, T., Fuchs, H.: Real-time projector tracking on complex geometry using ordinary imagery. In: Proc. CVPR, pp. 1–8 (2007)
5. Okatani, T., Deguchi, K.: Easy calibration of a multi-projector display system. *International Journal of Computer Vision* 85, 1–18 (2009)
6. Raj, A., Pollefeys, M.: Auto-calibration of multi-projector display walls. In: Proc. International Conference on Pattern Recognition (2004)
7. Cotting, D., Ziegler, R., Gross, M., Fuchs, H.: Adaptive instant displays: Continuously calibrated projections using per-pixel light control. *Computer Graphics Forum* 24, 705–714 (2005)
8. Zhou, J., Wang, L., Akbarzadeh, A., Yang, R.: Multi-projector display with continuous self-calibration. In: Proceedings of the 5th ACM/IEEE International Workshop on Projector camera systems, PROCAMS 2008, pp. 1–7. ACM, New York (2008)
9. Johnson, T., Welch, G., Fuchs, H., la Force, E., Towles, H.: A distributed cooperative framework for continuous multi-projector pose estimation. In: Proceedings of the 2009 IEEE Virtual Reality Conference, VR 2009, pp. 35–42. IEEE Computer Society, Washington, DC (2009)
10. Yang, R., Welch, G.: Automatic and continuous projector display surface calibration using every-day imagery. In: Skala, V. (ed.) Proceedings of Conference on WSCG (2001)
11. Zollmann, S., Langlotz, T., Bimber, O.: Passive-active geometric calibration for view-dependent projections onto arbitrary surfaces. *Journal of Visual Reality and Broadcasting* 4 (2007)
12. Okatani, T., Deguchi, K.: Autocalibration of a projector-camera system. *IEEE Trans. PAMI* 27, 1845–1855 (2005)
13. Triggs, B.: Autocalibration from planar scenes. In: Burkhardt, H.-J., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1406, p. 89. Springer, Heidelberg (1998)
14. Zach, C., Gallup, D., Frahm, J.M.: Fast gain-adaptive KLT tracking on the GPU. In: Proc. CVGP 2008, pp. 1–7 (2008)
15. Wu, C.: SiftGPU: A GPU implementation of scale invariant feature transform, <http://www.cs.unc.edu/~ccwu/siftgpu/>

3D Object Recognition Based on Canonical Angles between Shape Subspaces

Yosuke Igarashi and Kazuhiro Fukui

Graduate School of Systems and Information Engineering,
University of Tsukuba, Japan

igarashi@cvlab.cs.tsukuba.ac.jp, kfukui@cs.tsukuba.ac.jp

Abstract. We propose a method to measure similarity of shape for 3D objects using 3-dimensional shape subspaces produced by the factorization method. We establish an index of shape similarity by measuring the geometrical relation between two shape subspaces using canonical angles. The proposed similarity measure is invariant to camera rotation and object motion, since the shape subspace is invariant to these changes under affine projection. However, to obtain a meaningful similarity measure, we must solve the difficult problem that the shape subspace changes depending on the ordering of the feature points used for the factorization. To avoid this ambiguity, and to ensure that feature points are matched between two objects, we introduce a method for sorting the order of feature points by comparing the orthogonal projection matrices of two shape subspaces. The validity of the proposed method has been demonstrated through evaluation experiments with synthetic feature points and actual face images.

1 Introduction

In this paper, we propose a method to measure the similarity of 3D object shapes based on the geometrical relation between shape subspaces produced by the factorization method [1]. Using the proposed shape similarity measure, we realize 3D object recognition that is invariant to camera rotation and object motion.

The factorization method [1] is one of the most successful geometry-based methods for recovering the 3D shape of an object. The factorization method tracks the positions of multiple feature points through an image sequence and constructs a measurement matrix \mathbf{W} , which contains the 2D positions of the tracked feature points. The measurement matrix \mathbf{W} is then factored into the product of a motion matrix \mathbf{U} and a shape matrix \mathbf{V} . The motion matrix represents the camera rotation and the shape matrix represents the 3D positions of the object in a coordinate system attached to the object center.

The columns of the shape matrix span a 3-dimensional subspace, which is called the *shape subspace*. Shape subspace is invariant, under affine projection, to changes of coordinates caused by camera rotation and object motions [2, 3]. Therefore, the concept of shape subspace has been used in various tasks, such as motion segmentation [4, 5, 6] and sequential factorization [7]. This useful characteristic of shape subspaces leads us the idea that a shape similarity that is

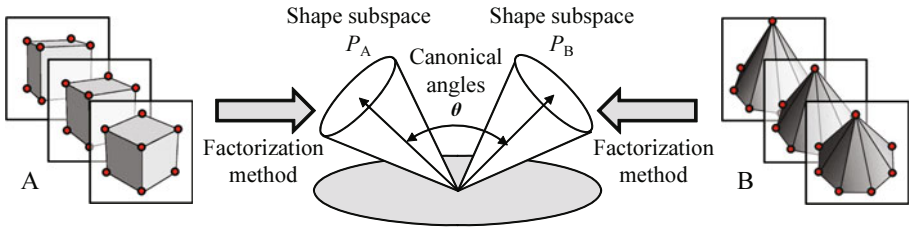


Fig. 1. Proposed framework of 3D object recognition based on canonical angles between shape subspaces

invariant to camera rotation and object motion can be established by measuring the geometrical relation between two shape subspaces. The shape subspace includes information about geometrical relations among multiple feature points. Therefore, we can obtain an index of the structural similarity between two sets of multiple feature points by measuring the canonical angles [8] between the two shape subspaces.

The usefulness of canonical angles (also called principal angles) has recently been established in applications in the field of computer vision, such as face recognition [9], where the relation between two subspaces representing distributions of face patterns is determined. Canonical angles have also been used for the motion segmentation of a non-ridge object [10], such as the human body. In this application, canonical angles are used to find the dimension of the intersection of two motion spaces that are produced by the factorization method. The dimension of the intersection indicates whether two parts are linked by a point or an axis.

Figure 1 shows the proposed framework for 3D object recognition. First, the feature points are tracked through image sequence for each object, and then the shape subspaces of the two objects are derived from the sets of the tracked feature points by the factorization method. Finally, the canonical angles between the shape subspaces are found and used to construct a measure of shape similarity. To obtain a robust measure of the similarity between shape subspaces, we have to overcome the problem that shape subspaces change depending on the order of the feature points used to construct a measurement matrix.

To do this, we use the concept of an orthogonal projection matrix, which is uniquely determined from the orthogonal basis vectors of a shape subspace. The core of our idea is to minimize the difference between the two orthogonal projection matrices, which are generated from the feature points of two objects, by rearranging the rows and the columns of one of them. The feature points are taken to have been matched between two objects when the difference between the two matrices is the smallest.

Several methods have been proposed for matching feature points based on shape subspaces. Wang and Xiao [11] applied QR factorization to the orthogonal projection matrices, and then permuted the rows in matrix \mathbf{Q} to produce a correspondence between shape subspaces. Marques and Costeira [12] used linear programming to compute a transformation matrix for minimizing the difference

between the orthogonal projection matrices. In this paper, we will compare the performance of the QR-based method with that of the proposed method, since both methods involve permuting a matrix.

The rest of the paper is organized as follows. Section 2 briefly describes the characteristics of shape subspaces. In Section 3, we propose the method for matching two sets of feature points and measuring shape similarity. In Section 4, we demonstrate the validity of the proposed method through experiments with a synthetic 3D object and images of real faces. Section 5 contains our conclusions.

2 Calculation Procedure of Shape Subspace

In this section, we outline how a shape subspace is generated. There are two calculation procedures: one is based on the factorization [1] of an image sequence, and the other is based on the positions of multiple feature points on an object.

2.1 Factorization of an Image Sequence

The factorization method [1] can robustly recover the shape and motion of an object from an image sequence without assuming a model of motion, such as constant translation or rotation. An image sequence can be represented as a $2F \times P$ measurement matrix \mathbf{W} , with P points tracked through F frames as follows:

$$\mathbf{W} = \begin{pmatrix} x_{11} & \cdots & x_{1P} \\ y_{11} & \cdots & y_{1P} \\ \vdots & \ddots & \vdots \\ x_{F1} & \cdots & x_{FP} \\ y_{F1} & \cdots & y_{FP} \end{pmatrix}, \quad (1)$$

where x_{fp} and y_{fp} are the 2D coordinates of the p th point in frame f .

If image coordinates are given with respect to their centroids, the measurement matrix \mathbf{W} is factored into the product of three matrices:

$$\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \simeq \mathbf{U}'\mathbf{\Sigma}'\mathbf{V}'^T, \quad (2)$$

where \mathbf{U} is a $2F \times 2F$ orthogonal matrix and \mathbf{V} is a $P \times P$ orthogonal matrix. $\mathbf{\Sigma}$ is a $2F \times P$ diagonal matrix with the singular values σ_i of \mathbf{W} in descending order. Here, the rank of matrix \mathbf{W} is 3 due to the geometrical constraint, so $\sigma_4, \dots, \sigma_D = 0$ (or are very small). Hence, \mathbf{W} can be represented as the product of a $2F \times 3$ matrix \mathbf{U}' , a 3×3 diagonal matrix $\mathbf{\Sigma}'$ and a $3 \times P$ matrix \mathbf{V}'^T as shown in Eq. (2).

The column vectors of the *shape matrix* \mathbf{V}' span the *shape subspace*. The shape subspace is invariant under an affine transformation of the set of feature points [4], such as that caused by camera rotation or object motion.

2.2 Generation Based on the Coordinates of Multiple Points

If the 3D coordinates of all the multiple feature points of an object are known, the shape subspace can be obtained directly without using the factorization method.

The shape subspace corresponding to an object is spanned by the column vectors of the $P \times 3$ matrix \mathbf{S} defined by

$$\mathbf{S} = (\mathbf{r}_1 \ \mathbf{r}_2 \ \dots \ \mathbf{r}_P)^T = \begin{pmatrix} x_1 & y_1 & z_1 \\ \vdots & \vdots & \vdots \\ x_P & y_P & z_P \end{pmatrix}, \tag{3}$$

where $\mathbf{r}_p = (x_p \ y_p \ z_p)^T$ for $1 \leq p \leq P$ denotes the positional vector of the p th point on an object. These vectors satisfy the relation $\sum_{p=1}^P \mathbf{r}_p = \mathbf{0}$. In this definition, the shape subspace is invariant to the selection of coordinates.

3 The Proposed Method

In this section, we first propose a method for matching feature points using an orthogonal projection matrix. Then, we explain how to measure the geometrical similarity between two shape subspaces using the canonical angles [8].

3.1 Matching Feature Points Using Orthogonal Projection Matrices

The shape subspace is the column space of \mathbf{V}' in Eq. (2) or of \mathbf{S} in Eq. (3). If the orders of feature points change, the shape subspace corresponding to them also changes. Therefore, we need to match the feature points between two objects to obtain shape similarity based on the geometric relation between shape subspaces.

The key property of the orthogonal projection matrix. The proposed method is based on the fact that an orthogonal projection matrix is uniquely determined by its corresponding object.

Let $\Phi = (\phi_1 \ \phi_2 \ \dots \ \phi_M)$ be an orthonormal basis for the M -dimensional subspace \mathcal{P} . The orthogonal projection matrix \mathbf{P} is then defined by

$$\mathbf{P} = \sum_{i=1}^M \phi_i \phi_i^T = \Phi \Phi^T. \tag{4}$$

Two shape matrices \mathbf{V}_A and \mathbf{V}_B obtained from the same object are not always equal, even if their feature points correspond to each other, because each shape matrix is just one set of basis vectors of the shape subspaces. Therefore, we cannot use the shape matrices to match the feature points. However, the two orthogonal projection matrices calculated from \mathbf{V}_A and \mathbf{V}_B using Eq. (4) always coincide:

$$\mathbf{Q} = \mathbf{V}_A \mathbf{V}_A^T = \mathbf{V}_B \mathbf{V}_B^T. \tag{5}$$

Based on this property, we match of feature points by rearranging the rows and columns of the orthogonal projection matrices corresponding to both objects, instead of handling the shape matrices.

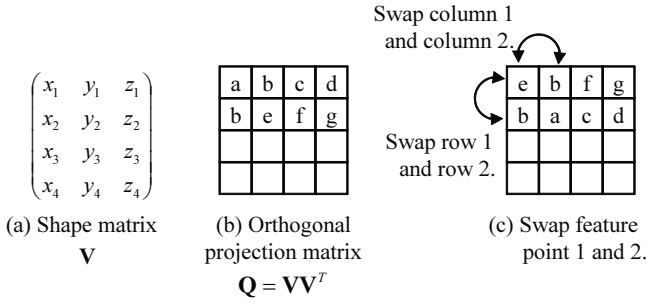


Fig. 2. Example of swapping rows and columns of the orthogonal projection matrix by swapping feature points ($P = 4$)

Exchanging feature points. Exchanging the order of two feature points on an object is equivalent to permuting the rows and columns of the orthogonal projection matrix. We will illustrate how to exchange the order of feature points by considering the following simple case.

Suppose that four feature points are extracted from an object. Figures 2 (a) and (b) show the shape matrix \mathbf{V} and the orthogonal projection matrix \mathbf{Q} calculated from \mathbf{V} . If feature point 1 and feature point 2 are exchanged, then the 1st row and the 2nd row are swapped in \mathbf{Q} , and the 1st column and the 2nd column are also exchanged at the same time, as shown in Fig. 2 (c). Note that the sets of the elements of the 1st row of (b) and the 2nd row of (c) are the same, although the orders of the elements are different. This rule is obeyed even if the number of the feature points to be exchanged increases.

Based on this rule, we can compare the rows of the orthogonal projection matrices by sorting the elements of the rows of each projection matrix in advance. The problem of matching feature points then reduces to finding the pairs of row vectors closest each other.

The Matching Algorithm. The procedure is as follows:

INPUT: $N \times N$ Orthogonal projection matrices \mathbf{X}_A and \mathbf{X}_B generated from N feature points of two objects A and B

OUTPUT: $N \times 2$ Correspondence matrix \mathbf{C}

1. **Initialization:** $\mathbf{Q}_{A(0)} = \mathbf{X}_A, \mathbf{Q}_{B(0)} = \mathbf{X}_B$
2. **for** $t = 0$ to N **do**
 - (a) Sort the unmasked elements of $\mathbf{Q}_{A(t)}$ and $\mathbf{Q}_{B(t)}$ within each row to produce temporary matrices $\mathbf{Q}'_{A(t)}$ and $\mathbf{Q}'_{B(t)}$.
 - (b) Find a pair of rows of $\mathbf{Q}'_{A(t)}$ and $\mathbf{Q}'_{B(t)}$ with the minimum L_1 -norm distance. The distance function between the row vectors, \mathbf{u}_i of $\mathbf{Q}'_{A(t)}$ and \mathbf{v}_j of $\mathbf{Q}'_{B(t)}$, is defined as follows:

$$d(\mathbf{u}_i, \mathbf{v}_j) = \sum_{k=1}^N |u_{ik} - v_{jk}| \quad (t = 0),$$

$$d(\mathbf{u}_i, \mathbf{v}_j) = \sum_{k=1}^{N-t} |u_{ik} - v_{jk}| + \sum_{k=1}^t |x_{ki}^* - y_{kj}^*| \quad (t \geq 1).$$

The row numbers found in the searching, r_A and r_B , are set to the t th row vector \mathbf{c}_t of \mathbf{C} , as $\mathbf{c}_t = (r_A, r_B)$.

- (c) Mask the r_A th row and the r_A th column $\mathbf{x}_{(t+1)}^*$ of $\mathbf{Q}_{A(t)}$, and the r_B th row and the r_B th column $\mathbf{y}_{(t+1)}^*$ of $\mathbf{Q}_{B(t)}$, respectively. These masked matrices are set to $\mathbf{Q}_{A(t+1)}$ and $\mathbf{Q}_{B(t+1)}$.

3. end for

Figure 3 shows a simple example of this matching procedure.

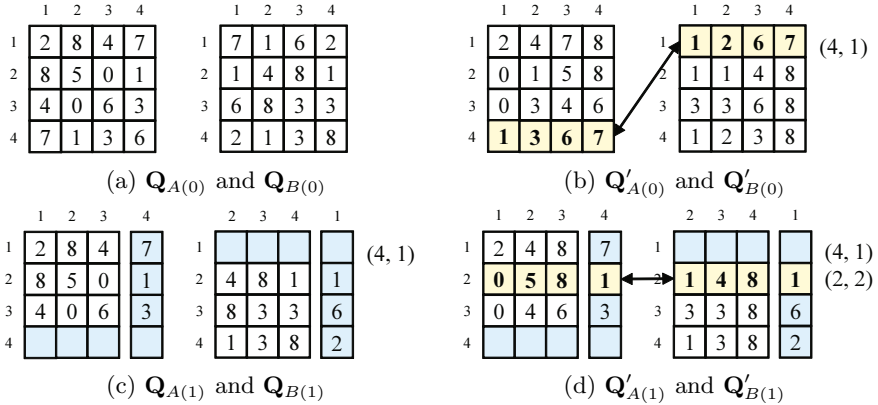


Fig. 3. Example of the proposed matching process. In this example, these matrices are not orthogonal projection matrices, although they are symmetric matrices. (a) shows the input matrices $\mathbf{Q}_{A(0)}$ and $\mathbf{Q}_{B(0)}$. In (b), the matrices are sorted within each row to produce temporary matrices $\mathbf{Q}'_{A(0)}$ and $\mathbf{Q}'_{B(0)}$. The 4th row of $\mathbf{Q}'_{A(0)}$ and the 1st row of $\mathbf{Q}'_{B(0)}$ are matched, as their L_1 -norm is the smallest. In (c), the 4th row of $\mathbf{Q}_{A(0)}$ and the 1st row of $\mathbf{Q}_{B(0)}$ are masked from the lists to be matched. Then, the 4th column and the 1st column are paired. These matrices are defined as $\mathbf{Q}_{A(1)}$ and $\mathbf{Q}_{B(1)}$. In (d), the non-corresponding elements of the rows of $\mathbf{Q}_{A(1)}$ and $\mathbf{Q}_{B(1)}$ are sorted. These matrices are $\mathbf{Q}'_{A(1)}$ and $\mathbf{Q}'_{B(1)}$. Then, the 2nd row of $\mathbf{Q}'_{A(1)}$ and the 2nd row of $\mathbf{Q}'_{B(1)}$ are matched.

3.2 Similarity between Shape Subspaces

First, we introduce canonical angles; then, we define the similarity between shape subspaces using them.

Consider an M -dimensional subspace \mathcal{S}_A and an N -dimensional subspace \mathcal{S}_B , where $M \leq N$. Given $\mathbf{u}_i \in \mathcal{S}_A$ and $\mathbf{v}_i \in \mathcal{S}_B$, the canonical angles θ_i ($\theta_1 \leq \theta_2 \leq \dots \leq \theta_M$) are uniquely defined by [8]

$$\cos^2 \theta_i = \sup_{\substack{\mathbf{u}_i \perp \mathbf{u}_j, \mathbf{v}_i \perp \mathbf{v}_j \\ 1 \leq i, j \leq M, i \neq j}} \frac{(\mathbf{u}_i \cdot \mathbf{v}_i)^2}{\|\mathbf{u}_i\|^2 \|\mathbf{v}_i\|^2}, \tag{6}$$

where (\cdot) denotes the inner product and $\|\cdot\|$ denotes the norm of a vector.

Let \mathbf{Q}_A and \mathbf{Q}_B denote the orthogonal projection matrices of the subspaces \mathcal{S}_A and \mathcal{S}_B . Then, $\cos^2 \theta$ for the canonical angle θ between \mathcal{S}_A and \mathcal{S}_B is equal to the eigenvalue of $\mathbf{Q}_A \mathbf{Q}_B$ or $\mathbf{Q}_B \mathbf{Q}_A$ [8]. The largest eigenvalue corresponds to the

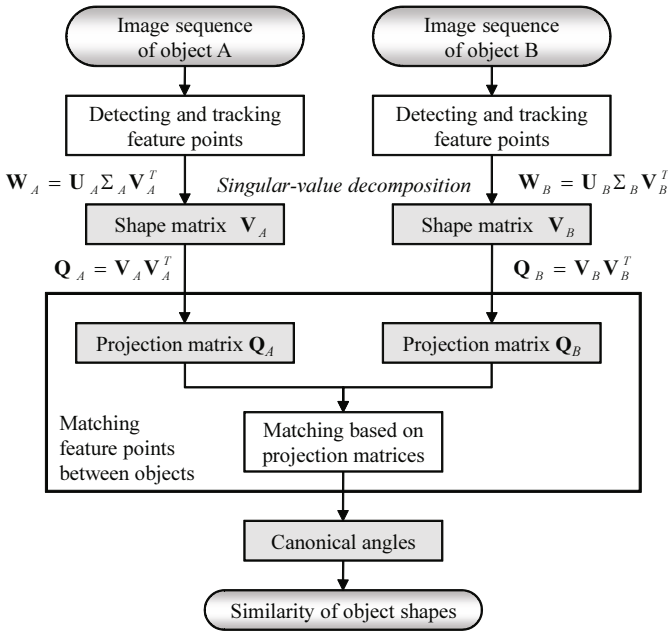


Fig. 4. Flow of the object recognition process based on the proposed similarity measure

smallest canonical angle θ_1 , whereas the second largest eigenvalue corresponds to the smallest angle θ_2 in a direction perpendicular to that of θ_1 . The values of $\cos^2\theta_i$ ($i = 3, \dots, M$, and $M \leq N$) are calculated similarly.

From these canonical angles, we define the shape similarity φ by

$$\varphi = \frac{1}{M} \sum_{i=1}^M \cos^2\theta_i. \tag{7}$$

If two shape subspaces coincide completely with each other, φ is 1.0, since all canonical angles are zero. The similarity φ gets smaller as the two spaces separate. Finally, the similarity φ is zero when the two subspaces are orthogonal to each other.

3.3 3D Object Recognition Based on the Proposed Similarity Measure

Figure 4 shows the proposed procedure, from inputting the image sequences of two objects A and B to the output of the shape similarity index.

First, multiple feature points are tracked through an image sequence of object A by a tracker, such as the Kanade-Lucas-Tomasi (KLT) feature tracker [13]. Then, the measurement matrix \mathbf{W}_A is calculated from the positions of the tracked feature points. Next, the measurement matrix \mathbf{W}_A is factored into the product of the shape matrix \mathbf{V}_A and the motion matrix \mathbf{U}_A . A shape matrix \mathbf{V}_B

and a motion matrix \mathbf{U}_B are also obtained from the image sequence for object B. The orthogonal projection matrices \mathbf{Q}_A and \mathbf{Q}_B are calculated from \mathbf{V}_A and \mathbf{V}_B . Their rows and columns are rearranged to match feature points. Then, the shape similarity φ can be calculated from the shape subspaces using Eq. (7).

4 Experimental Results

In this section, we first use synthetic data to evaluate the accuracy of the proposed algorithm for matching feature points, and then use images of real faces to demonstrate the effectiveness of the proposed method for object recognition.

4.1 Experiment I: Matching Feature Points Using Synthetic Data

We evaluate the robustness of the proposed matching method using a synthetic 3-dimensional data set. We prepared two sets of feature points for the evaluation experiment. The first set is a set of P randomly generated points on a unit sphere. The second set is the first set with added Gaussian noise of standard deviation σ . Two shape matrices were generated from both the sets of feature points using Eq. (3) in Sec. 2.2. We compared the proposed matching method with the matching method based on QR factorization [11] described in Sec. 1.

Figure 5 shows an example of feature-point matching for $P = 30$ and $\sigma = 0.1$. Figure 6 shows a comparison of the error rates of the two methods of matching for various values of the parameters P and σ . For each of the parameters, 200 independent experiments were run. The proposed method consistently shows a lower error rate than the QR based method [11]. When $\sigma = 0.1$ and $P = 30$, the error rate is about 20% (see Fig. 6 (a)). If $P = 100$ and $\sigma = 0.0316$, the error rate with our method is about 5% (see Fig. 6 (b)). We conclude that the proposed matching method has high accuracy and is robust even under high noise conditions.

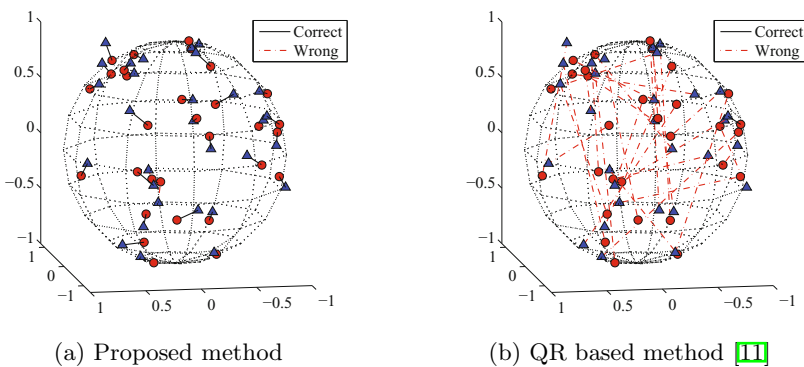


Fig. 5. Example of matched feature points on spheres ($P = 30, \sigma = 0.1$)

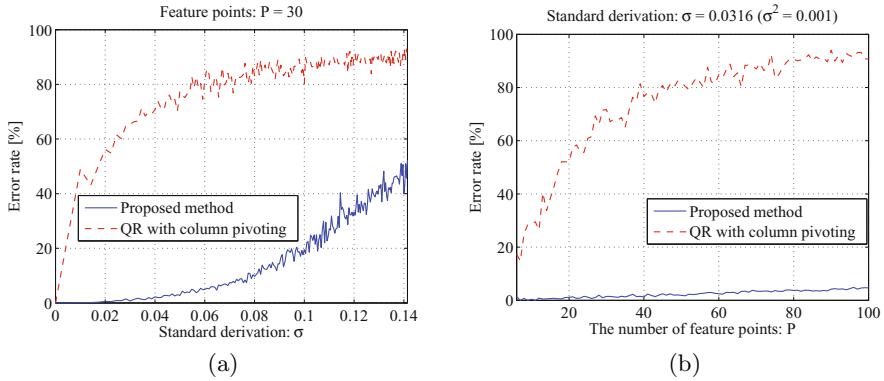


Fig. 6. Performances of the proposed matching method: (a) error rate vs. level of noise ($P = 30$) and (b) error rate vs. the number of feature points ($\sigma = 0.0316$)

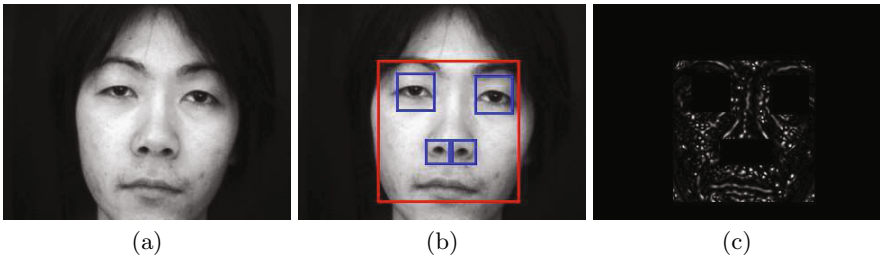


Fig. 7. Pre-processing for generating the shape subspace: (a) input image, (b) detected face, pupils and nostrils, (c) separability map

4.2 Experiment II: Face Recognition

We now consider the application of the proposed method to face recognition. The surface of a human face has many feature points, such as moles and freckles, which are distinct characteristics that can be used to identify individuals. The effectiveness of using these feature points for face recognition has been shown by Pierrard and Vetter [14]. We detected moles and freckles from facial images using a circular separability filter [15], and used them as feature points.

The number of participants was 22. A participant sat on a chair about 1 meter away from a camera. We captured 300 frames for each participant, while the head was moving. The image size was 1024×768 pixels.

Figure 7 shows examples of the input image, detected face region and separability map. First, we detected the facial region [16] and the regions of pupils and nostrils [15]; we then remove the latter regions from the facial region, because they are common features of all subjects. Next, we applied a circular separability filter to obtain a separability map. Finally, we detected and tracked 26 feature points from the 300 separability maps by applying the KLT feature tracker [13]. Figure 8 shows examples of the tracked feature points.

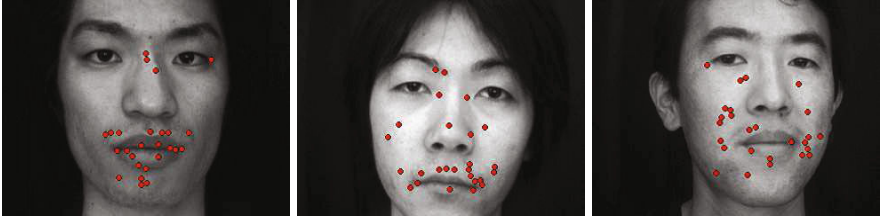


Fig. 8. Examples of detected and tracked feature points ($P = 26$)

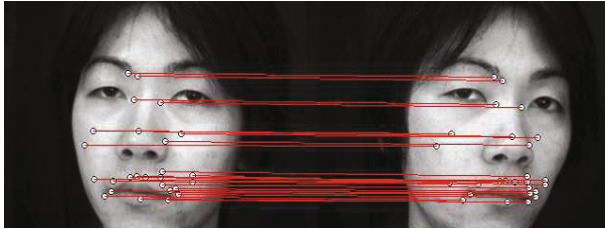


Fig. 9. An example of feature points matched between two image sequences

The 300 frames were divided into sets of 30 frames for each of the 22 subjects so that we obtained 220 datasets. A shape subspace was generated from each dataset by the factorization method. We compared the proposed matching method and the conventional matching method using QR factorization in terms of classification performance. The input subspace generated from a set of input image sequences was classified using the Nearest Neighbor algorithm. The classification rate was estimated by the Leave-One-Out method.

Figure 9 shows an example of the feature points matched. Figure 10 shows the similarity maps among the sets of sequential images by the proposed method. Figure 10 (a) shows the result by the proposed matching method and (b) shows that by the conventional method. Table 1 lists the recognition rates and Equal Error Rate (EER), which is defined as the crossing point of the False Acceptance Rate and False Rejection Rate curves. The value of ERR should be as low as possible to achieve high performance face recognition.

From Table 1 we can see that the proposed matching method is superior to the conventional, QR-based method. The recognition rate of the proposed method was 99.5% with 22 subjects whereas the recognition rate using the QR-based method was 94.1%. The large difference between the performances of the two methods seems to derive from the degree of robustness of feature extraction against ambiguity resulting from added noise and occlusions. Moreover, the EER of the proposed method is very low: it is only 2.60%, compared to 17.3% for the QR-based method. These results clearly support the validity of our framework for 3D object recognition based on the canonical angles between shape subspaces.

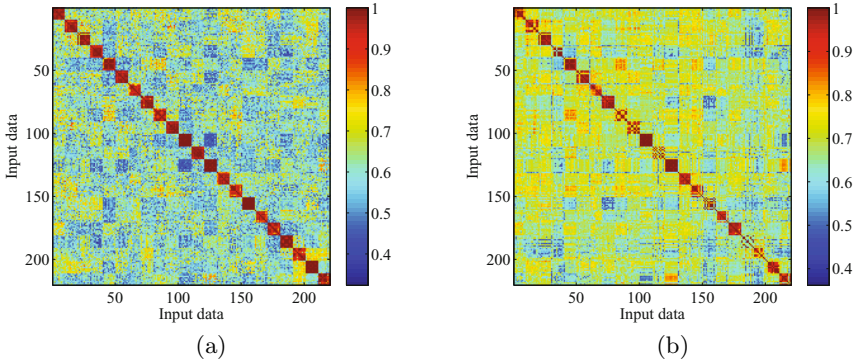


Fig. 10. Similarity maps based on the canonical angles for face recognition with 22 subjects: (a) using the proposed matching method and (b) using the conventional method based on QR factorization [11]

Table 1. Comparison between the proposed method and the conventional method for face recognition

Matching method	Recognition rate	EER
Proposed	99.5%(219/220)	2.60%
QR-based [11]	94.1%(207/220)	17.30%

5 Conclusions

In this paper, we have proposed a method for measuring the similarity between 3D object shapes, which is invariant to camera rotation and object motion. The proposed measure of shape similarity is based on the shape subspaces produced by the factorization method. The shape subspace produced depends on the order of the feature points considered. To avoid this ambiguity, we have proposed a method of matching the feature points of two objects by rearranging the rows and columns of their orthogonal projection matrices.

We have confirmed through an evaluation experiment using synthetic data that the proposed matching method can match the feature points of two objects. Our method is more robust to noise than the conventional method based on QR factorization. We have also demonstrated that a framework based on the combination of shape similarity and our matching method is effective for classifying facial images.

References

1. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision* 9, 137–154 (1992)

2. Poelman, C.J., Kanade, T.: A paraperspective factorization method for shape and motion recovery. *Pattern Analysis and Machine Intelligence* 19, 206–218 (1997)
3. Begelfor, E., Werman, M.: Affine invariance revisited. In: *Computer Vision and Pattern Recognition*, vol. 2, pp. 2087–2094 (2006)
4. Costeira, J.P., Kanade, T.: A multi-body factorization method for independently moving objects. *International Journal of Computer Vision* 29, 159–179 (1998)
5. Ichimura, N.: Motion segmentation based on factorization method and discriminant criterion. In: *International Conference on Computer Vision*, vol. 1, pp. 600–605 (1999)
6. Kanatani, K.: Motion segmentation by subspace separation and model selection. In: *International Conference on Computer Vision*, vol. 2, pp. 586–591 (2001)
7. Morita, T., Kanade, T.: A sequential factorization method for recovering shape and motion from image streams. *Pattern Analysis and Machine Intelligence* 19, 856–867 (1997)
8. Chatelin, F.: *Eigenvalues of Matrices*. John Wiley & Sons, Chichester (1993)
9. Fukui, K., Yamaguchi, O.: Face recognition using multi-viewpoint patterns for robot vision. In: *International Symposium of Robotics Research*, pp. 192–201 (2003)
10. Yan, J., Pollefeys, M.: A factorization-based approach for articulated nonrigid shape, motion and kinematic chain recovery from video. *Pattern Analysis and Machine Intelligence* 30, 867–877 (2008)
11. Wang, Z., Xiao, H.: Dimension-free affine shape matching through subspace invariance. In: *Computer Vision and Pattern Recognition*, pp. 2482–2487 (2009)
12. Marques, M., Costeira, J.: Lamp: Linear approach for matching points. In: *International Conference on Image Processing*, pp. 2113–2116 (2009)
13. Tomasi, C., Kanade, T.: Detection and tracking of point features. *Carnegie Mellon University Technical Report CMU-CS-91-132* (1991)
14. Pierrard, J.S., Vetter, T.: Skin detail analysis for face recognition. In: *Computer Vision and Pattern Recognition*, pp. 1–8 (2007)
15. Fukui, K., Yamaguchi, O.: Facial feature point extraction method based on combination of shape extraction and pattern matching. *Systems and Computers in Japan* 29, 49–58 (1998)
16. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: *Computer Vision and Pattern Recognition*, pp. 511–518 (2001)

An Unsupervised Framework for Action Recognition Using *Actemes*

Kaustubh Kulkarni¹, Edmond Boyer¹, Radu Horaud¹, and Amit Kale²

¹ INRIA, Grenoble, Rhone Alpes

² Siemens Corporate Technology, Bangalore

{firstname.lastname}@inria.fr, kale.amit@siemens.com

Abstract. In speech recognition, phonemes have demonstrated their efficacy to model the words of a language. While they are well defined for languages, their extension to human actions is not straightforward. In this paper, we study such an extension and propose an unsupervised framework to find phoneme-like units for actions, which we call *actemes*, using 3D data and without any prior assumptions. To this purpose, build on an earlier proposed framework in speech literature to automatically find actemes in the training data. We experimentally show that actions defined in terms of actemes and actions defined by whole units give similar recognition results. We define actions out of the training set in terms of these actemes to see whether the actemes generalize to unseen actions. The results show that although the acteme definitions of the actions are not always semantically meaningful, they yield optimal recognition accuracy and constitute a promising direction of research for action modeling.

1 Introduction

Recognition of human actions is an important part of research in dynamic scene understanding. The applications of classifying human actions in a video extend from video indexing and retrieval, video surveillance, human-robot and human-computer interactions. There are several challenges which arise while tackling the problem of human action recognition. One such fundamental problem is the temporal representation of actions. Phonemes in spoken language are the smallest or distinct segmental unit of sound which can be combined or concatenated to form words. This fact is exploited in speech recognition where Hidden Markov Models (HMMs) are learned on these phonemes. These models are combined to define the words of a vocabulary. Motivated from speech recognition, we investigate whether such a hierarchical definition is possible for human actions using sub-action units which we call *actemes*.

Intuitively, there must exist a restricted set of generic motions of a human body which can define all actions. This set, if it exists, can be likened to a set of phonemes which can define every word in the dictionary from a given language. There are certain advantages if such actemes can be learned. Firstly, the actemes would allow us to define a large number of actions in a compact representation. Secondly, the advantage of having a hierarchy i.e. where an action is described as sequence of actemes is that when a new action is added to the list of actions

to be recognized, this new action can be described in terms of actemes thus obviating the need for learning a new model every time an action is added.

The concept of phoneme definition for words exist from linguists. No such widely accepted definitions of actemes exist for human actions. Several researchers have tried to come up with such definitions. Green et. al. [1] proposes the use of 35 *Dynemes* which form the basic units of human actions or skills. The *dynemes* are defined in terms joint angles. An HMM model is used for action recognition. Another work by [2] defines *kinetemes* on the joint angle space of human motion. These *kinetemes* form the basic unit of a human activity language. Using these *kinetemes* and language grammar like rules the authors propose to construct any complex human action. Bregler [3] defines *Movemes* as linear dynamical systems over which an HMM model is learned for recognition. Since the space of all possible human motions is very large and since no widely acceptable definition exist it is better to automatically come up with these definitions for actemes as opposed to [1]. Also, we assume no rules while labelling the actions in terms of the learned actemes, as done in [2] instead we use the recognition algorithm it self to provide the labelling. In [3], the author proposes a method to automatically learn the *Movemes* from the training set. The results shown in this paper are evaluated on actions consisting on repeated segments such as walking, running and skipping. In such a scenario the basic blocks constructing the actions are obvious and eliminate the need for labelling the actions in terms of *Movemes*. In this paper, for the experiments we evaluate the efficacy of our proposed method exclusively on actions which do not consist of repeated segments.

In this paper, we build on a speech recognition formalism [4,5], which proposes to design a recognizer terms of acoustic subword units (ASWU). This method assumes no prior information while learning the ASWUs from words. It learns these definitions in an unsupervised data-driven manner. We apply this method from speech recognition for obtaining actemes because it is completely data-driven and makes no prior assumptions on the definitions of actemes. This is a completely different way to approach the problem of human action representation and recognition than the earlier proposed methods. Secondly, the number of actemes per action is also known so a data-driven approach is best suited to come up with acteme units. To summarize, the main contributions of the paper are the following: (1) We use a speech recognition formalism to learn the actemes and the representation of actions in terms of the actemes in an unsupervised framework. (2) We show that actions from outside the training set can be represented in terms of these learned actemes and recognized without explicitly learning a new model for the actions.

2 Related Work

Automatic annotation of actions in videos is a challenging task and various action recognition methods can be grouped together depending on the types of features used and the method employed to model the temporal and spatial representations of actions. A brief survey of temporal representations similar to

ours has been discussed in Sec. 1. We restrict our survey to 2D silhouettes, 3D visual hulls and key frames as features. The recognition methods discussed are HMMs and dynamic time warping (DTW) based methods. For a detailed survey of action recognition methods see [6].

The earliest features used were silhouettes extracted from each frame over time and an HMM was learned from them [7]. These were used to recognize tennis strokes from single views. A later paper [8] describes temporal templates for human action recognition. [9] extends the 2D temporal templates to 3D volumes. [10] also describes a view invariant recognition method where they learn parametric HMMs from 3D data and use the HMMs as a generative model to synthesize 2D action sequences closest to an unknown 2D test action sequence. Another way of classifying actions is by using dynamic time warping (DTW). [11] learns the warping bounds for the actions from the training data. [12] proposes to use distance between linear dynamical systems for action classification. [13], [14] perform action recognition by defining actions as trajectories on the Grasmann Steifel manifold. [15] extends the DTW framework using average templates with multiple features to model intra-class variances and perform simultaneous recognition and localization of actions in a video sequence. All these methods learn the model on entire actions.

Another popular method is to define actions as a set of poses or key frames or exemplars [16]. They use single key frames to recognize backhand and forehand in tennis. There also has been work which uses short snippets of frames [17] to recognize actions instead of a single frame. In [18], the authors use the forward selection algorithm to find the most discriminative set of exemplars to describe an action vocabulary. [19] model actions as a sequence of atomic body poses where the authors consider the order in which the poses appeared. In this paper, we express action in terms of sequence of short segments or actemes instead of sequence of key poses.

3 The Method

To learn the actemes we employ a method proposed in speech recognition [4, 5]. Actemes are equivalent to phonemes or ASWUs and the whole actions are equivalent to a word. In this method, the authors propose to 1) optimal cut the words into piecewise stationary segments, 2) get a reduced set of ASWUs by applying K-means on the means of each optimal segment, 3) learn HMMs on these ASWU, 4) apply the connected word Viterbi algorithm to label the training data in terms these ASWU to generate a lexicon or a phonetic definition for each word in terms of the ASWUs, and 5) then use this definition in the Viterbi framework to perform recognition. Instead of using HMMs to model the actemes we use the earlier proposed average template models [15] and the one-pass dynamic programming algorithm [20] for labeling and the modified one-pass dynamic programming algorithm [21] for recognition. The average template model is shown to outperform the HMMs in [15]. Figure 1 and Figure 2, explains the building blocks of the algorithm. These building blocks are explained in the following sub-sections from 3.1 to 3.3.

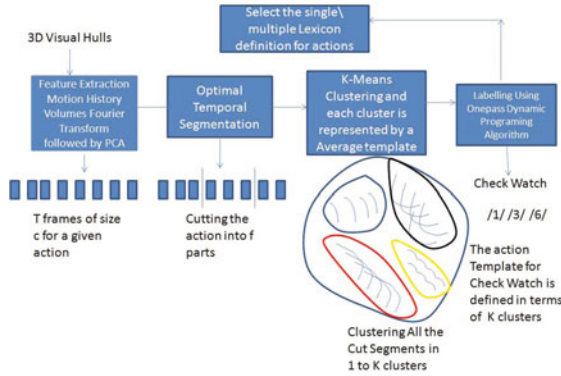


Fig. 1. In this figure, we have the block diagram of the acteme training steps

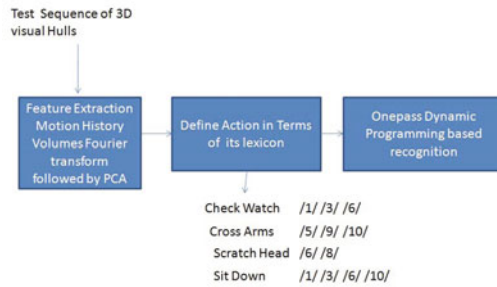


Fig. 2. In this figure, we have the block diagram of the steps for recognition

Feature Computation. The features needed to interface with a time synchronous onepass-DP algorithm should be a set of feature vectors over time given by $E = \{e_1, e_2, \dots, e_b, \dots, e_T\}$ where e_b is a vector of dimension c at a given time instant b . In this paper, we use 3D visual hulls as our features. We then compute motion history volumes (MHVs) [9] to recognize action using DTW. The MHVs store the motion history on a 3D occupancy grid in a given window. In this paper, we use a short window of size 5. The occupancy grid of the MHVs is of the size $64 \times 64 \times 64$ for each frame. Since the actors are allowed to change their view point freely we convert the Cartesian coordinates to cylindrical coordinates followed by Fourier transform on this occupancy grid. This Fourier magnitudes, of size $16 \times 16 \times 16$, are invariant to the rotation around the z-axis. We perform further dimensionality reduction using principal component analysis to reduce the feature vector to a size of 100. Therefore, in this paper we have $c = 100$.

Temporal Segmentation. Several approaches using HMMs have been used for action, gesture and sign language recognition. The implicit assumption of using a left to right HMMs for recognition is that the action is composed of piecewise stationary regions. These regions are modelled by the states of the HMM. Hence the number of states is an important parameter to correctly estimate for action recognition. The piecewise stationary regions in a word are the phonemes and if

we fit a left to right HMM model with the number of states equal to the number of phonemes we get a good recognition accuracy [5]. Also the steady state regions are most likely to lie between abrupt motion changes or discontinuities which can be used for temporal segmentation [22] using MHVs. We do not take the approach of using velocity discontinuities because actions like "stand" or "sit down" do not have abrupt changes in the direction of velocity. We motivate our strategy to cut actions into relevant regions by assuming that the actions can be decomposed into piecewise stationary regions. The method cuts the actions into segments such that the global distortion of these segments w.r.t their means is minimized. This can be formulated as a dynamic programming problem [23].

Consider an action template defined as set of features over time by E where e_b is a feature vector corresponding to the b th frame of size $c = 100$ and the action is performed over T time instances. The task here is to segment E into f homogeneous segments by minimizing the sum of the distances between frames of the segments to their respective means. Let the segment boundaries for a given action template be $G = \{g_1, g_2, g_i, \dots, g_f\}$ where g_i are integers indicating the frame numbers of the boundaries. The i^{th} segment starts at $g_{i-1} + 1$ and ends at g_i ; $g_1 = 0$ and $g_f = T$. The optimal boundaries G^* can be found by minimizing the following function over all possible segmentations:

$$D_1(f, T) = \sum_{i=1}^{i=f} \sum_{b=g_{i-1}+1}^{g_i} d_1(e_b, \bar{e}_i) \tag{1}$$

where $D_1(f, T)$ is the total accumulated distance for segmenting E into f segments. The mean of the i th segment is given by \bar{e}_i which is the average of the frames of the i th segment given by $H = \{e_{g_{i-1}+1}, \dots, e_{g_i}\}$. The distance metric used is euclidean; $d_1(e_b, \mu_i) = \|e_b - \bar{e}_i\|$.

The problem of solving for optimal boundaries can be efficiently solved using a trellis realization. This can be achieved by solving the following dynamic programming recursions as given in [23], [24]:

$$D_1(i, g_i) = \min_{g_{i-1}} [D_1(i - 1, g_{i-1}) + d_1(e_b, \bar{e}_i)] \text{ where } b = g_{i-1} + 1 \text{ to } g_i \tag{2}$$

where $D_1(i, g_i)$ is the cost of dividing the template E into i segments till the frame g_i where $i < f$. This cost is given by the minimum over cost accumulated by dividing E into $i - 1$ segments till frame g_{i-1} plus the distance of the i th segment with its mean. The optimal segmentation can be found by backtracking through the trellis starting from $\min D(f, T)$.

If the number of segments f for a given word is equal to the number of phonemes in that particular word then ASWUs are equivalent to phonemes of that language otherwise the ASWUs are not semantically meaningful. The number of phonemes in a given word is not always known because of the pronunciation. In [4,5], it is shown that even if ASWUs are not semantically meaningful the algorithm still provides a good recognition accuracy. Since the number of actemes in an actions are unknown the method given in [4,5] is more suited to be applied to the problem of action recognition using actemes as opposed to other approaches [1,2,3] motivated from speech recognition systems.

Clustering and Computing average-template Model. This procedure to segment each action template into f segments is repeated for all actions in the training set. Therefore, if there are N training instances of all actions then we will have a set of $f \times N$ variable length temporal segments. To get a compact representation we apply K-means on this set of temporal segments to get the K actemes. Since, we assumed that each segments is piecewise stationary we represent each segment in this set by its mean and apply the K-means on the these segment means.

To represent the cluster corresponding to each of the acteme we compute a temporal average or nominal template [25] over all the instances of a given acteme. In this section, we describe a method to represent each acteme as an average of the templates in that cluster of actemes. The average pattern or average-template R^k is computed by mapping the segments, $H = \{H_1, H_2, \dots, H_l, \dots\}$, in the cluster corresponding to the acteme k using DTW. We use Euclidean distance as the local distance $d_2(i, j)$ between the frame i of R^k and frame j of H . If I is the length of R^k and J is the length of H , the path is forced to begin at the point $D_2(1, 1)$ and end at $D_2(I, J)$ on the trellis to compute the accumulated distance $D_2(i, j)$. This accumulated distance is defined as:

$$\begin{aligned}
 D_2(i, j) = \min[& D_2(i - 2, j - 1) + 3d_2(i, j), \\
 & D_2(i - 1, j - 1) + 2d_2(i, j), \\
 & D_2(i - 1, j - 2) + 3d_2(i, j)]
 \end{aligned} \tag{3}$$

where i is the frame index of the average reference pattern R^k and j is the frame index of the train pattern H .

Backtracking from the point $D_2(I, J)$ on the trellis yields the optimal path $p = [i_m, j_m]$ and the corresponding mapped set of feature vectors $[R^k(i_m), H(j_m)]$. Here m , is the index of a point on the optimal path p . The average reference pattern R_l^k for an activity is computed by the successive weighted averaging of l instances as follows:

$$R_l^k(m) = \left(1 - \frac{1}{l}\right) R_{l-1}^k(i_m) + \frac{1}{l} H_l(j_m), m = 1 \dots M \tag{4}$$

where M is the number of points on the optimal path p and $R_{l-1}^k(i_m)$ is the average of the previous $l - 1$ templates. The new time axis for the instance R_l^k is computed as:

$$p_1(m) = \left(1 - \frac{1}{l}\right) i_m + \frac{1}{l} j_m, m = 1 \dots M \tag{5}$$

We linearly transform this new time axis to a constant length P where P is the average length of all segments in the cluster of acteme k . The transformation is done as follows:

$$p_2(m) = \frac{P}{M} p_1(m) \tag{6}$$

as $p_2(m)$ would have non-integer values we define a time axis $p_3(m')$ where $m' = 1, 2, 3 \dots P$. The feature values of the average pattern $R_l^k(m)$ are interpolated to

get the new average pattern representing the cluster corresponding to acteme k $R_l^k(m')$.

Labelling. In this section, we discuss the method to label each of the training sequences in terms of the learned K learned actemes. We use a 'connected word recognition' algorithm based on the one-pass DP, well known in speech recognition [20]. Continuous labelling of action templates in terms of actemes is a difficult task to do on line, primarily because this involves the problem of jointly determining the optimal number of actemes M^* in the train sequence \mathbf{O} , their boundaries $S^* = \{s_0^*, s_1^*, s_{m-1}^*, s_m^*, \dots, s_{M^*}^*\}$ and associated optimal acteme indices $I^* = \{i_1^*, i_2^*, \dots, i_m^*, \dots, i_{M^*}^*\}$ (where $v_{i_m^*} \in V$), by minimizing a measure of distance $D(\mathbf{O}, \mathbf{R})$ between the train sequence \mathbf{O} and a typical reference acteme template sequence $\mathbf{R} = \{R_{v_{i_1}}, R_{v_{i_2}}, \dots, R_{v_{i_m}}, \dots, R_{v_{i_M}}\}$ each drawn from V . The decoding problem of determining (M^*, S^*, I^*) is solved by minimizing $D(\mathbf{O}, \mathbf{R})$ over the variables (M, B, I) using the time-synchronous one-pass DP decoding algorithm.

To compute the optimal cumulative distance, we use two types of transition rules (a) for acteme interior i.e. Within Acteme Recursion (b) for acteme boundary i.e Cross-Acteme Recursion. These recursions are computed for all frames of the train action template w.r.t the all frames of all average template acteme models in a left to right time synchronous manner. These recursions would then result in many possible paths. The optimal action sequence or path will be the one which corresponds to the minimum cumulative distance (Termination and Backtracking).

We now provide the mathematical details pertaining to the above intuitive explanation of the algorithm. The acteme vocabulary of size K is given by $V = \{v_1, v_2, \dots, v_K\}$. Each acteme corresponds to a reference pattern $R_{v_k}(k')$, where $k' = 1, 2, 3 \dots P_{v_k}$; P_{v_k} is the number of frames of the average template v_k^{th} acteme where $k = 1, 2, 3 \dots K$. The train action template frame index is given by q and Q is the length of the train action template \mathbf{O} . During the labelling pass the sequence of warping is given by the average-templates. The local distance between one frame of the average template of a given acteme and a frame of the training action sequence is computed in the following way:

$$d(q, k', v) = \|R_v(k') - O(q)\| \tag{7}$$

Let D denote the global accumulated distance between the train action frame and the reference pattern frame. The one-pass DP decoding would look to minimize the global accumulated distance over all the frames of the train action pattern. The following steps give a method to accumulate the global distance between a given train action frame and a frame of the reference pattern to find a globally optimal path:

1. **Within acteme recursion:** This recursion is computed for all frames Q of the train action pattern and all frames k' of all reference patterns except for $k' = 1$ i.e. the recursions are applied to to all frames except at the acteme beginning. This recursion can be denoted as:

$$D(q, k', v) = d(q, k', v) + \min_{k'-2 \leq r \leq k'} (D(q-1, r, v)) \tag{8}$$

2. **Cross-acteme Recursions:** This recursion is computed for all Q test frames and for $k' = 1$ frames of all reference patterns. This recursion allows a transition into the first frame of a given reference pattern from the last frame of all other reference pattern including the given reference pattern or it allows the path to be in the last frame of that given reference pattern i.e. the algorithm either stays in the particular acteme or transits into the first frame on any other acteme depending on which of the two paths yields a minimum score. It can be denoted as:

$$D(q, 1, v) = d(q, 1, v) + \min_{1 \leq v \leq K} [D(q-1, P_v, v), D(q-1, 1, v)] \quad (9)$$

3. **Termination and Backtracking:** To find the best acteme sequence the algorithm uses the following termination condition at the train action frame Q :

$$D^* = \min_{1 \leq v \leq K} [D(Q, P_v, v)] \quad (10)$$

The algorithm checks for the minimum accumulated distance for the best path at the last frame of every reference pattern at the train action frame Q . The best path is backtracked from that point through back-pointers stored during the Within Acteme and Cross Acteme recursions.

The output of running the onepass-DP algorithm will be a sequence of optimal acteme indices I^* for every training action template. For eg. the a given sequence \mathbf{O} could be labelled as $\{v_3, v_2, v_7\}$. This is a completely unsupervised labelling by an onepass DP algorithm which is also the same algorithm we use for recognition. We choose the acteme representation or lexicon which repeats itself the most number of times as the model for a given action while recognition. Since, there is intra class variance in the manner in which different actors perform the action we find that upto 4 lexicons have to be used to get results close to our baseline. This is true in the case of speech recognition where a given word can be pronounced by different by different speakers one phonetic representation is not enough to obtain good recognition results. The lexicons chosen are in descending order of their occurrence while labelling the training data.

Recognition. While recognizing the actions we assume that the action boundaries in the video sequence are known. Therefore, we only recognize the action and do not localize it in the video sequence. This assumption is necessary as isolated action recognition is the true test of the efficacy of this approach as it gives only substitution errors i.e. the an action can be recognized as itself or confused as some other action. Simultaneous recognition and localization causes insertion and deletion errors.

We use the method proposed in [21] to perform action recognition when each action is defined in terms of actemes. The proposed algorithm can be used for simultaneous recognition and localization of action in a video sequence. We switch off the *Cross-Word transitions* [21] since we are only recognizing the actions and assume that the boundaries in the video sequence are known.

Let the number of action to be recognized be $W = \{w_1, w_2, \dots, w_m, \dots, w_M\}$ where $m = 1 \dots M$ is the total number of actions in the recognition vocabulary.

The number of lexicons per actions is defined as l which is the constant and same for every action to be recognized in the vocabulary. The l^{th} representation of each action w_m in terms of the actemes given by $\{a_{1m}^l, a_{2m}^l, \dots, a_{jm}^l, \dots, a_{N_{am}}^l\}$ where $j = 1 \dots N_{am}$ is the number of actemes representing the action w_m and $l = 1 \dots L$. The local distance between the test sequence and the average templates be $d(m, a_{jm}^l, l, k', q)$ where $k' = 1, 2, 3 \dots P_{a_{jm}^l}$; $P_{a_{jm}^l}$ is the number of frames of the average template a_{jm}^l acteme. The test action sequence consist of $q = 1 \dots q \dots Q$ frames. The local distance is given by euclidean distance between the k^{th} frame of the average template representing each acteme and the q^{th} frame of the test data. The dynamic time warping time synchronously calculates the minimum global accumulated distance $D(m, a_{jm}^l, l, k', q)$ to reach the k^{th} frame of the word w_m represented by lexicon l till the q^{th} of the test action sequence. Since, the cross action recursions are switched off there are only with action recursions which can be divided into two types:

1. **Within Acteme recursions:** These recursions are applied for all frames of the average template of each of the acteme except for the template beginning i.e. $k' = 1$

$$D(m, a_{jm}^l, l, k', q) = d(m, a_{jm}^l, l, k', q) + \min_{k' - 2 \leq r \leq k'} [D(m, a_{jm}^l, l, r, q - 1)] \quad (11)$$

2. **Cross Acteme recursions:** The actions are represented by variable number of actemes in a given order. This order is given in the labelling step. Therefore, in this recursion the transition occurs into the first frame of the average template of each acteme from the last frame of the average template of the previous acteme. This is known as forced alignment of the concatenated action model to the test sequence. This recursion are applied at the first frame $k' = 1$ of every acteme except the first. This step can be mathematically denoted as:

$$D(m, a_{jm}^l, l, k' = 1, q) = \min [D(m, a_{jm}^l, l, k' = 1, q - 1), \quad (12)$$

$$D(m, a_{(j-1)m}^l, l, P_{a_{(j-1)m}^l}, q - 1)]$$

where $j = 2$ to N_{am}

3. **Termination and Backtracking:** The action is assumed to begin at the first frame of the average template of the first acteme of the given action and end at the last frame of the last acteme of the same action. Therefore, the optimal accumulated distance D^* can be obtained by checking the last frames of the last actemes of all actions with all representative lexicons at the last frame Q of the test sequence:

$$D^* = \min_{m=1 \dots M} \min_{l=1 \dots L} D(m, a_{N_{am}}^l, l, P_{a_{N_{am}}^l}, Q) \quad (13)$$

The test sequence is classified as the action index m for which the D is minimum.

4 Experimental Results

In this section, we show that the actions described as actemes give equivalent performance to the actions when modelled as whole units themselves. We add actions which are not included in the training set to check whether the learned actemes generalize to unseen data. We evaluate our method on the INRIA XMAS dataset. In our experiments, we assume that the boundaries of the actions in the video sequences are known.

For the first set of experiments, we use a reduced vocabulary of *check watch, sit down, get up, punch and kick*. The recognition experiments are performed on the set of 10 actors and are validated by the standard leave-one-out testing procedure. In these experiments, we show that the actions described by actemes give equivalent recognition performance to actions described as whole units themselves. We first obtain the 100 dimensional feature vectors in time from the 3D visual hulls using the procedure described in Sec. 3. We apply the temporal segmentation procedure on all available training instances of each actions to get 2 and 3 segments respectively. If the method tries to cut the actions into more than three segments we observe that the actions start breaking into segments which are only 1-2 frames long. These segments cannot be averaged with longer segments to learn an average template model because the warping of very short with long segments is meaningless 5.

We apply K-means on these cut segments and plot the recognition results with K varying from 10 to 50 in the steps of 10. Due to the intra class variance observed in the performance of the actions we increase the number of lexicons per actions from 1 to 4 in the descending order of their occurrence. We find that both help in increasing the recognition accuracy. The recognition results of the first set of experiments are given in Fig. 3. We observe two facts from this result:

1. For the case where training actions are cut into 2 segments the recognition accuracy increases till two lexicons and then it starts to decrease. This is because there is trade off between the number of lexicons per actions and the recognition accuracy as increasing the number of lexicons per actions also increases the possibility of confusions.
2. Recognition accuracy is better when the actions are segmented into 3 parts than 2 parts because if we observe the reduced vocabulary of actions apart for *stand up and sit down* in the XMAS dataset they consist of three parts

Table 1. Comparison of actemes representation with other Recognition Methods

Added Action	LDA 9	PCA 9	Mahalanobis 9	Average Template 15	<i>Actemes</i> This Paper	Average Trajectory 11
<i>None</i>	94.67	86.67	95.33	95.33	94.00	92.00
<i>CrossArms</i>	97.78	81.67	97.79	97.79	87.29	86.74
<i>ScratchHead</i>	92.22	77.22	93.33	97.78	81.67	91.11
<i>PickUp</i>	96.67	83.89	94.44	97.24	91.71	92.82

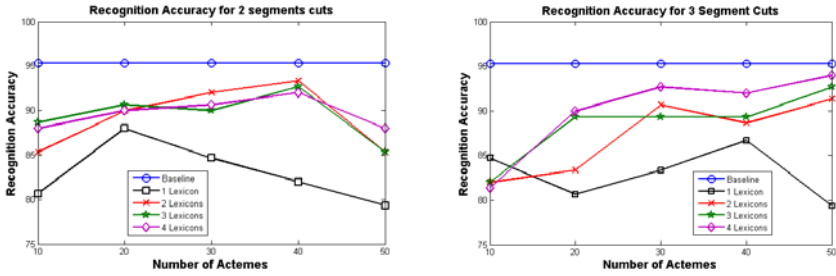


Fig. 3. Recognition accuracy plots for 2 and 3 segments cuts. We can see that the recognition accuracy of acteme based representation is close to the baseline of actions when modelled as whole units themselves.

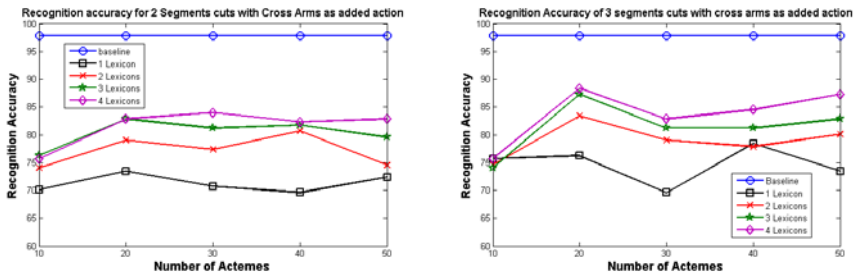


Fig. 4. Recognition accuracy plots for 2 and 3 segments cuts with *Cross Arms* as the added action. We can see that for the 3 cut case this 6 word vocabulary gives satisfactory results.

an initial movement, the main action and the relaxation part. In *sit down*, there is an initial movement of bending the back, crossing the legs sitting down and coming to a relaxed pose after sitting down. *Stand up* is exact opposite of sitting down.

In the second set of experiments in the paper, we add the following set of actions one at a time to the *cross arms*, *scratch head*, *pick up* to the list of actions to be recognized. Thus, forming a vocabulary of 6 actions every time one of the 3 actions is added. The training instances of these actions are not used to train the actemes. We only use the training instances from these actions to get the lexical representation of the added action in terms of the actemes learned from the earlier 5 actions.

We observe that the recognition accuracy is the best for *pick up* because the initial part of *pick up* is very similar to the *sit down* and the latter part of pick is similar to the *stand up* action. Therefore, the actemes for *pick up* are present in the reduced vocabulary of 5 actions. The next best performance is achieved by *cross arms* again because the action *check watch* is similar to it. The actions *scratch head* action when added to the vocabulary of 5 action gives the worst recognition result because there are no actions in the reduced vocabulary of 5

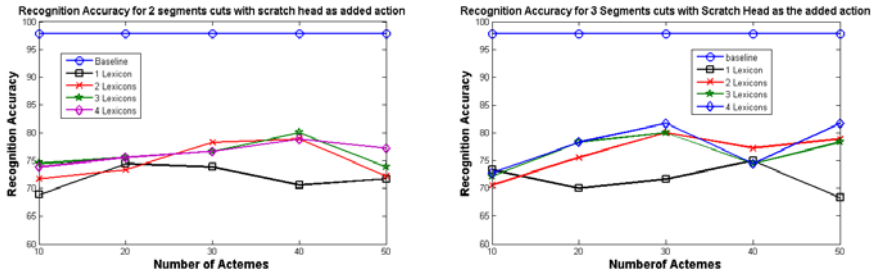


Fig. 5. Recognition accuracy plots for 2 and 3 segments cuts with *Scratch Head* as the added action. We observe that the recognition results are poor because the *scratch head* does not have a similar action in reduced vocabulary of 5 actions used to learn the actemes.

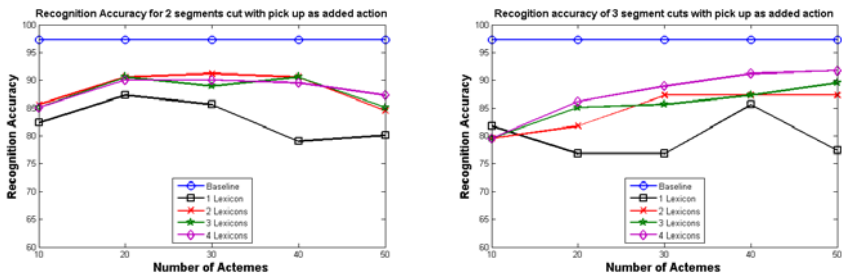


Fig. 6. Recognition accuracy plots for 2 and 3 segments cuts with *Pick Up* as the added action. We observe that the recognition results are good because the actemes for pick up are present in the *Stand Up* and *Sit Down* action.

actions similar to the *scratch head* action. The recognition results for the second set of experiments are given in Fig. 4 to Fig. 6.

We compare our results with other methods which were applied to the INRIA XMAS database. We model these whole units of actions using an average template model as discussed in [3] and perform isolated action recognition [15] to get the recognition baseline. This baseline is shown as the blue baseline in Fig. 3 to Fig. 6. We also compare our results with the recognition method proposed in [9]. The size of the FFT features used to obtain recognition results $16 \times 16 \times 16$. We compare our method with the approach in [11] which proposes another method to learn average or nominal trajectories. The average trajectories are computed using the 100 dimensional features described in Sec. 3. We find that the proposed acteme based representation performs slightly better than the method proposed in [11] and comes close to the recognition performance of [15, 9] for the first set of experiments using 5 actions. For the second set of experiments we find that for added actions, *cross arms*, *pick up*, which have a similar action in the 5 action training set the results are comparable to all the baselines. The recognition accuracy in the *actemes* column is the recognition accuracy achieved for 3 cuts and $K = 50$. All the methods use leave-one-out testing strategy. [9]

uses a best segment representative of the action. While actemes, [15] and [11] use the boundaries extracted from the ground truth.

In the experimental section, we have discussed the efficacy of the actemes w.r.t. the recognition accuracy. The representation of *check watch and sit down* actions in terms of the actemes is shown in the video uploaded with the paper.

5 Conclusion and Future Work

To conclude, we have demonstrated an unsupervised framework to learn a set of actemes from a given training database to represent actions. We experimentally show that actions defined in terms of these actemes can give the similar recognition accuracy as compared to the whole unit themselves. We also showed that satisfactory recognition results can be achieved even with action which are not included in the training set for learning the actemes. For future work we would like to explore techniques which can be semi-supervised to learn semantically meaningful actemes. We would also like to extend this framework to bag-of-words like approaches. The next obvious step would be to do simultaneous recognition and localization of actions in a video sequence.

Acknowledgements. This work was done under European project *HUMAVIPS* (FP-ICT 2009 247525). The authors would like to thank Pavan Kumar Turaga, Center for Automation Research, Univ. of Maryland for providing the code to compute the FFT features. We thank Dr. V. Ramasubramanian for his time and helpful insights on speech recognition algorithms. Our sincere thanks to Ashok Veeraraghvan for sharing with us his code from [11].

References

1. Green, R.D., Guan, L.: Quantifying and recognizing human movement patterns from monocular video images-part i: a new framework for modeling human motion. *IEEE Trans. Circuits Syst. Video Techn.* 14, 179–190 (2004)
2. Guerra-Filho, G., Aloimonos, Y.: A language for human action. *Computer* 40, 42–51 (2007)
3. Bregler, C.: Learning and recognizing human dynamics in video sequences. In: *CVPR* (1997)
4. Lee, C.H., Soong, F., Juang, B.H.: A segment model based approach to speech recognition. In: *ICASSP* (1988)
5. Rabiner, L., Juang, B.: *Fundamentals of Speech Recognition*. Prentice-Hall, New Jersey (1993)
6. Poppe, R.: A survey on vision-based human action recognition. *Image and Vision Computing* 28, 976–990 (2010)
7. Yamato, J., Ohya, J., Ishii, K.: Recognizing human action in time-sequential images using hidden markov models. In: *CVPR*, pp. 379–385 (1992)
8. Bobick, A.F., Davis, J.W.: The recognition of human movement using temporal templates. *PAMI* (2001)
9. Weinland, D., Ronfard, R., Boyer, E.: Free viewpoint action recognition using motion history volumes. In: *CVIU*, vol. 104, pp. 249–257 (2006)

10. Weinland, D., Boyer, E., Ronfard, R.: Action recognition from arbitrary views using 3d exemplars. In: ICCV (2007)
11. Veeraraghavan, A., Chellappa, R., Roy-Chowdhury, A.K.: The function space of an activity. In: CVPR (2006)
12. Turaga, P.K., Veeraraghavan, A., Chellappa, R.: From videos to verbs: Mining videos for events using a cascade of dynamical systems. In: CVPR (2007)
13. Turaga, P.K., Veeraraghavan, A., Chellappa, R.: Statistical analysis on stiefel and grassmann manifolds with applications in computer vision. In: CVPR (2008)
14. Turaga, P.K., Chellappa, R.: Locally time-invariant models of human activities using trajectories on the grassmanian. In: CVPR (2009)
15. Kulkarni, K., Cherla, S., Kale, A., Ramasubramanian, V.: A framework for indexing human actions in video. In: ECCV Workshops (2008)
16. Carlsson, S., Sullivan, J.: Action recognition by shape matching to key frames. In: CVPR Workshops (2001)
17. Schindler, K., Gool, L.V.: Action snippets: How many frames does human action recognition require? In: CVPR (2008)
18. Weinland, D., Boyer, E.: Action recognition using exemplar-based embedding. In: CVPR (2008)
19. Ogale, A.S., Karapurkar, A., Aloimonos, Y.: View-invariant modeling and recognition of human actions using grammars. In: ICCV Workshops (2005)
20. Ney, H.: The use of one-stage dynamic programming algorithm for connected word recognition. *IEEE Trans. on Acoustic Speech and Signal Processing* 32(2), 263–270 (1984)
21. Ramasubramanian, V., Kulkarni, K., Kaemmerer, B.: Acoustic modeling by phoneme templates and modified one-pass dp decoding for continuous speech recognition. In: ICASSP (2008)
22. Weinland, D., Ronfard, R., Boyer, E.: Automatic discovery of action taxonomies from multiple views. In: CVPR (2006)
23. Svendsen, T., Soong, F.: On the automatic segmentation of speech signals (1987)
24. Ramasubramanian, V., Sreenivas, T.: Automatically derived units for segment vocoders. In: ICASSP, vol. 1, pp. I-473–I-476 (2004)
25. Zelinski, R., Class, F.: A learning procedure for speaker-dependent word recognition systems based on sequential processing of input tokens. In: ICASSP (1983)

Segmentation of Brain Tumors in Multi-parametric MR Images via Robust Statistic Information Propagation

Hongming Li, Ming Song, and Yong Fan

National Laboratory of Pattern Recognition, Institute of Automation
Chinese Academy of Sciences, Beijing, China, 100190
{hml,msong,yfan}@nlpr.ia.ac.cn

Abstract. A method is presented to segment brain tumors in multi-parametric MR images via robustly propagating reliable statistical tumor information which is extracted from training tumor images using a support vector machine (SVM) classification method. The propagation of reliable statistical tumor information is implemented using a graph theoretic approach to achieve tumor segmentation with local and global consistency. To limit information propagation between image voxels of different properties, image boundary information is used in conjunction with image intensity similarity and anatomical spatial proximity to define weights of graph edges. The proposed method has been applied to 3D multi-parametric MR images with tumors of different sizes and locations. Quantitative comparison results with state-of-the-art methods indicate that our method can achieve competitive tumor segmentation performance.

1 Introduction

Reliable segmentation of brain tumors from MR images is of great importance for surgical planning and therapy assessing. Diligent efforts have been made to achieve time-efficient, accurate, and reproducible tumor segmentation. It however remains a challenging task to achieve robust segmentation as brain tumors differ much in appearance, location, size, and shape.

Many methods have been proposed for tumor image segmentation in the literature, including supervised classification methods, unsupervised clustering methods, and active contour methods. Supervised classification methods [8,10,14,16] perform image segmentation using classifiers built on training data with assumption that the statistical information extracted by the classifiers from the training data can cover testing data. Their performance therefore relies on the consistency between the training and the testing data. Due to imaging noise and patient anatomy difference, discrepancy inevitably exists between the training and testing data, which often leads to degraded segmentation performance. Rather than relying on the training data, unsupervised methods [1,6,9,13,15,17,18,26] perform segmentation by partitioning the image to be segmented using its specific

intensity information. These methods may alleviate the problem of image intensity variability; however they often require an appropriate number of clusters to be assigned to achieve a good performance. Unlike aforementioned methods, the active contour methods perform image segmentation utilizing both image intensity and geometrical information of objects to be segmented [2,7,12,24,28].

Rather than performing tumor segmentation within the existing frameworks, we propose a fully automatic method by utilizing reliable statistical tumor information obtained from a support vector machine (SVM) classifier to guide a graph theory based tumor segmentation. The key elements of the proposed approach are: 1) a statistical model is built upon training images with labeled tumors using SVM to provide reliable statistical tumor information for images to be segmented [4]; 2) a graph theoretic semi-supervised learning approach is utilized to propagate the reliable statistical tumor information to all the image space with local and global consistency [27]; 3) a robust “edge stopping” function is adopted to embed image boundary information in the graph edge weight measurement for limiting information propagation between image voxels of different properties [3]. The proposed method has been applied to brain tumor segmentation of 3D multi-parametric MR images. Quantitative experiment results indicate that our method can achieve promising segmentation performance. Extensive validation experiments also demonstrate our method’s robustness to its parameters.

2 Methods

The tumor segmentation is implemented as a semi-supervised learning problem with guidance of reliable statistical tumor information obtained from a SVM classifier built on available training data.

2.1 Graph Theoretic Approach for Semi-supervised Segmentation

Graph theory based segmentation approaches model the image to be segmented as a graph $G(V, E)$ where each node of V corresponds to a voxel of the image and each edge of E connects a pair of voxels and is associated with a weight of pair-wise voxel similarity. With the graph theory based image representation, the image segmentation problem is solved by assigning different labels to graph nodes and can be performed by methods like graph cut and random walks [11,22,25]. To best utilize statistical tumor information of training data and alleviate problems of inter-image intensity variability, we adopt a semi-supervised graph theoretic approach that is able to propagate labeling information of a small number of graph nodes to unlabeled nodes with local and global consistency [27].

Given labeling information of a small number of graph nodes, labels of the graph’s nodes can be predicted by exploiting the consistency between nodes based on the cluster assumption that nearby nodes or nodes on the same structure are likely to have the same label [5,27]. The labeling problem can be solved by minimizing a cost function within a regularization framework [27]:

$$Q(F) = F^T(I - S)F + \mu(F - L_{ini})^T(F - L_{ini}) \quad (1)$$

where I is an identity matrix and S is the normalized edge weight matrix, F is the segmentation label vector and L_{ini} is the initial label vector. The first term of Eq. (II) is a local consistency constraint to encourage nearby nodes to have similar labels, and the second term measures the consistency between the labeling result and the initial labeling information. These two terms are balanced by the parameter μ to achieve a labeling with local and global consistency. The minimization of $Q(F)$ can be achieved using an iterative procedure which has been demonstrated to converge to the optimal solution [27]:

$$F^{m+1} = (1 - \alpha)SF^m + \alpha L_{ini} \tag{2}$$

where F^k is the updated label information at the k -th iteration, F^0 is equivalent to the L_{ini} , $0 < \alpha < 1$ is a parameter related to μ , trading off the information from the initial labeling and the prediction results. This iterative procedure can be regarded as label information propagation. At each iteration, every node absorbs the label information from other nodes and retains partial label information of its initial state. The label information is updated until convergence and each node is assigned to the class from which it receives the most information.

It is worth noting that label information of nodes is updated by the spread of label information of other nodes according to their corresponding edge weights. For a successful segmentation it is critical to get properly defined edge weights and a few reliably labeled nodes.

2.2 Robust Edge Weight Measurement

As the propagation based learning strategy achieves the labeling via spreading the available labeling information according to pair-wise edge weights, the edge weight measurement plays an important role in the segmentation. Typically, only the image intensity similarity and spatial proximity are taken into account in the edge weight measurement [22]. However, in tumor MR images, the overlap between the intensity range of healthy tissues and that of tumors always exists and the locations of tumors vary much. The image intensity information and spatial proximity might not be able to distinguish tumor from healthy tissues very well. To mitigate this problem, we incorporate image boundary information into the edge weight computation:

$$w_{ij} = e_{ij}^I \times e_{ij}^L \times e_{ij}^g \tag{3}$$

where e_{ij}^I and e_{ij}^L are measures of image intensity similarity and spatial proximity, e_{ij}^g is an image boundary information term, i and j are different nodes in the graph.

The image similarity and spatial proximity terms e_{ij}^I and e_{ij}^L are defined as [22]:

$$e_{ij}^I = e^{-\|F_i - F_j\|^2 / \sigma_F^2} \tag{4}$$

$$e_{ij}^L = \begin{cases} e^{-\|L_i - L_j\|^2 / \sigma_L^2} & \text{if } \|L_i - L_j\| < r \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where F_k refers to the image intensity vector of the voxel (node) k , L_k is the spatial location of the voxel (node) k , σ_F and σ_L are free parameters controlling scales of the kernels. The neighborhood size of each node is controlled by the parameter r , edge weight is set to 0 for any pair of nodes that are more than r apart.

The image boundary information is embedded in an “edge stopping” function which could be any monotonically decreasing function to make it robust to image noises [3]. In particular, we use a function based on Turkey’s biweight robust estimator for embedding image gradient information between nodes and the image boundary information term is defined as:

$$e_{ij}^g = \begin{cases} \frac{1}{2}[1 - (G_{ij}/\sigma_g)^2]^2 & \text{if } G_{ij} \leq \sigma_g \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

where G_{ij} is the maximum image gradient magnitude along the i - j direction between voxels i and j , and σ_g is a free parameter controlling the spatial scale of the function. The gradient magnitude of images with a vector value at each voxel is calculated as the difference between the maximal and minimal eigenvalues in a principle component analysis of the partial derivatives as described in [21]. The value of σ_g can be estimated using robust statistics [3]. This term works as an indicator to the presence of an image boundary between voxels i and j . A small value of e_{ij}^g means the probability that voxels i and j are located in the same region is low and the information propagation between them should be limited. The e_{ij}^g term makes the parameter selection in the edge weight calculation more stable as it tries to constrain the information propagation between nodes from different objects. The “edge stopping” function involves the computation of image gradient and searching for the maximum gradient along the line inbetween voxels i and j within the neighborhood.

2.3 Label Initialization Using Reliable Statistical Information

To get guidance information, i.e., a number of labeled voxels, we adopt a supervised classification method that has been shown capable of achieving promising tumor segmentation performance [14]. However, due to the discrepancy between training and testing data, the supervised classification segmentation method does not work well for testing data not well covered by the training information. Therefore, we select voxels with the most reliable classification results to initialize our graph theory based segmentation.

To build a classifier for tumor segmentation, a SVM based strategy is adopted [14]. In particular, a support vector machine (SVM) classifier with probabilistic outputs is built on intensity information of multi-parametric images and the training data are obtained from labeled tumor images [4, 14]. Each voxel of a multi-parametric image contains vector valued intensity information. Elements of the vector valued intensity information are scaled to be distributed in $[0, 1]$ separately after multi-parametric images are spatially aligned and their bias fields are corrected [14]. The intensity normalization is implemented globally and does not change the relative contrast of tumors in the image. For each voxel, the

feature vector used in classification consists of image intensity information of all voxels in its spatial neighborhood [14]. Gaussian radial basis function kernel is used in the SVM classification and the classification parameters are tuned using cross-validation.

When applied to testing images, the SVM classifier provides each voxel a label indicating tumor or healthy tissue and a probability measure indicating the reliability of the classification. Based on the probability measure, we select voxels (nodes) with tumor or healthy tissue probability measure higher than a threshold as the candidates for the label initialization. In particular, small connected regions containing only a small number of voxels are abandoned. In order to enhance the reliability of the initial labeling, outliers are further excluded from the candidate set. The outliers are voxels whose intensities are far from the robust means estimated from the candidate samples using the Minimum Covariance Determinant estimator [20] for tumor and normal tissues respectively. All the remaining candidates are selected as the initial labels.

According to this label initialization, the tumor segmentation is obtained by propagating the reliable statistical tumor information to all other unlabeled nodes in the graph based on the edge weight defined in Section 2.2. The main procedure of our method is summarized as:

1. A SVM classifier is built on the training data of multi-parametric MR images with both tumor and healthy tissues.
2. Testing multi-parametric MR image is segmented using the SVM classifier, and initial label information is determined using the selection procedure described above.
3. Construct a weighted graph based on the image to be segmented.
4. Iterate $F^{m+1} = (1 - \alpha)SF^m + \alpha L_{ini}$ until convergence.

3 Experimental Results

The method is validated on both UCINIA simulated brain tumor MR images [19] and real MR images with tumors of different sizes and locations.

3.1 Evaluation on UCINIA Simulated MR Images

Five subjects are available in the dataset with tumor segmentation ground truth, each of them having T1-weighted, T2-weighted and contrast enhanced T1-weighted MR images. Non-brain parts of these images are removed prior to the segmentation using the mask generated based on the probability map of brain tissues available in the dataset. However, the non-brain parts of head images can also be removed using publicly available software packages in conjunction with manual editing.

The image data of one subject is selected as the training data for building a SVM classifier, and other subjects are used as the testing data. The image intensities of the testing images are globally scaled to have a similar distribution to the training data using a histogram-match method, which is accomplished by mapping intensities through intensity cumulative distribution function (CDF)

of the source image and the CDF of the training image. Spatial neighborhood with a size of $3 \times 3 \times 3$ is used to get features for the SVM classification and the parameters of the SVM classifier are optimized by a 5 fold cross-validation. The constructed SVM classifier is then applied to the testing images and classification results are used for label initialization. In particular, voxels with healthy tissue probability higher than 0.999 are treated as label initialization candidates. As the tumor probability from the SVM classification varies much due to the inter-subject image intensity variability, instead of setting a hard threshold for the tumor label initialization we choose a number of voxels with the highest tumor probability as the candidates. The label initialization is achieved through the robust selection process on these candidates, and the reliable label information is then propagated to all other voxels to obtain the final segmentation. All parameters of the algorithm remain unchanged for all the testing subjects. The size of the spatial neighborhood for graph construction is $5 \times 5 \times 5$.

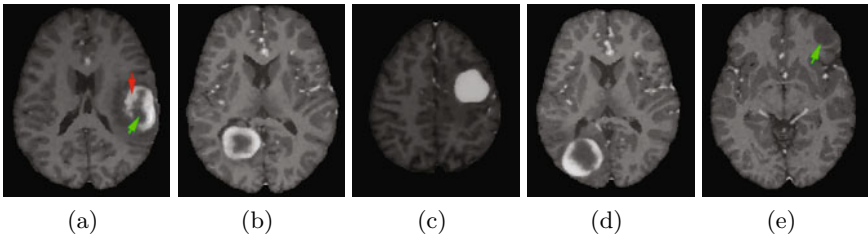


Fig. 1. One slice of the contrast enhanced T1-weighted image for each subject (a to e). Subject (a) is used as the training data, and the SVM based segmentation for subject (e) fails.

The SVM based segmentation method fails to segment one testing subject due to the fact that the tumor region in this subject's contrast enhanced T1-weighted image has similar intensities to cerebrospinal fluid (CSF), different from the training data as shown in Fig. 1. The SVM classifier using only image intensity as features does not work in this case in that training-testing image intensities are not matched. The classifier is sensitive to the enhanced part of tumor indicated by the red arrow and insensitive to the non-enhanced part indicated by the green arrow in Fig. 1a. Therefore, the tumor region in Fig. 1e (indicated by the green arrow) cannot be detected. However, our graph based algorithm can successfully segment the tumor with several manually selected tumor and normal tissue voxels. To make a fair comparison among different automatic segmentation algorithms, we focus on the other subjects for validating our tumor segmentation method.

Besides the SVM classification method for tumor segmentation [14], we compare our method with a hidden Markov random field based segmentation method, implemented by FAST [17, 26]. To test if it is helpful to add the image boundary information term in the graph edge weight measurement, we also perform the graph based segmentation with edge weight computed with only image

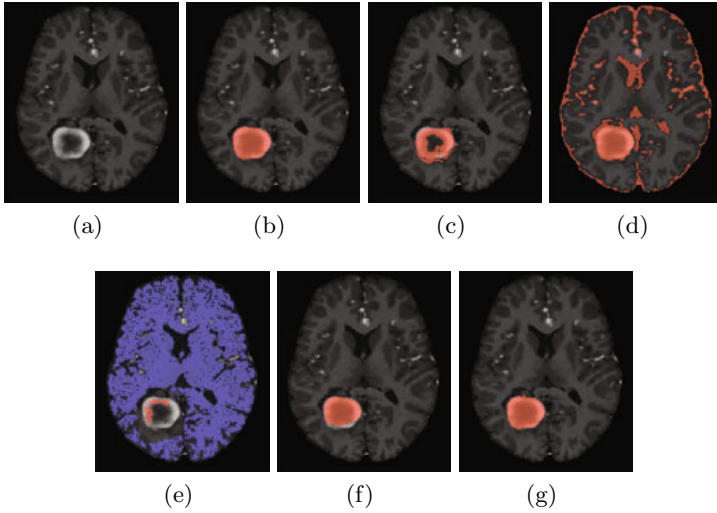


Fig. 2. One slice of a testing simulate image, ground truth, and segmentation results by different methods. (a) the original contrast enhanced T1-weighted image, (b) the ground truth, (c) segmentation obtained by the SVM classification, (d) segmentation using FAST, (e) the initial labeled image for information propagation, (f) segmentation using information propagation method without gradient information used in edge weight measurement, and (g) segmentation of our method.

Table 1. Mean and standard deviation of Jaccard similarity on simulate MR images

Method	Mean	Standard deviation
SVM based segmentation	0.604	0.136
FAST	0.086	0.026
Traditional information propagation	0.777	0.044
Our method	0.937	0.008

intensity similarity and spatial proximity and refer to this method as traditional information propagation.

Fig. 2 shows a representative slice of one testing subject’s contrast enhanced T1-weighted image, tumor segmentation ground truth, and its associated tumor segmentation results obtained by methods to be compared (top row, a~d; bottom row, f), as well as tumor segmentation results at different stages of our method (bottom row, e and g). As shown in Fig. 2c, the SVM classifier cannot successfully detect the boundary and the necrotic region of the tumor, which might be due to the fact that the insufficient training on only one training subject cannot well cover the tumor intensity distribution with high variability. Fig. 2d shows the segmentation result of FAST with the class number set to be 4 [26]. As FAST is a model driven technique which requires a good estimation of number of classes to be segmented even for normal brain tissue segmentation (grey matter, white matter and CSF), we try different numbers of tissue

classes including 2, 4, 5 respectively with the purpose of achieving segmentation with different meanings (2 for tumor and non-tumor; 4 for grey matter, white matter, CSF, and tumor; 5 for grey matter, white matter, CSF, tumor, and other); however, the tumor region cannot be separated well from other brain tissues in any settings. It is worth noting that the best performance is shown in Fig.2f, which shows that the tumor and CSF could not be distinguished due to the high similarity in their intensity information. However, with the guidance of reliably labeled voxels obtained from the SVM classification, the graph theory based information propagation achieves better performance, as shown in Fig.2f and Fig.2g. By comparing results shown in Fig.2f and Fig.2g, it can be found that the segmentation with boundary information is more robust to the tumor boundary. In Fig.2f the segmentation of traditional information propagation may be confused at the boundary by wrongly label information interchange due to that the edge similarity computed using only multi-parametric intensities and spatial positions cannot robustly distinguish tumor voxels from normal ones.

Besides visual inspection, we also use Jaccard similarity to quantitatively compare the segmentation results. The Jaccard similarity is the normalized intersection in voxel space of two segmentations, i.e., $(X \cap Y)/(X \cup Y)$ (automatic segmentation result X and ground truth Y). The Jaccard similarity is

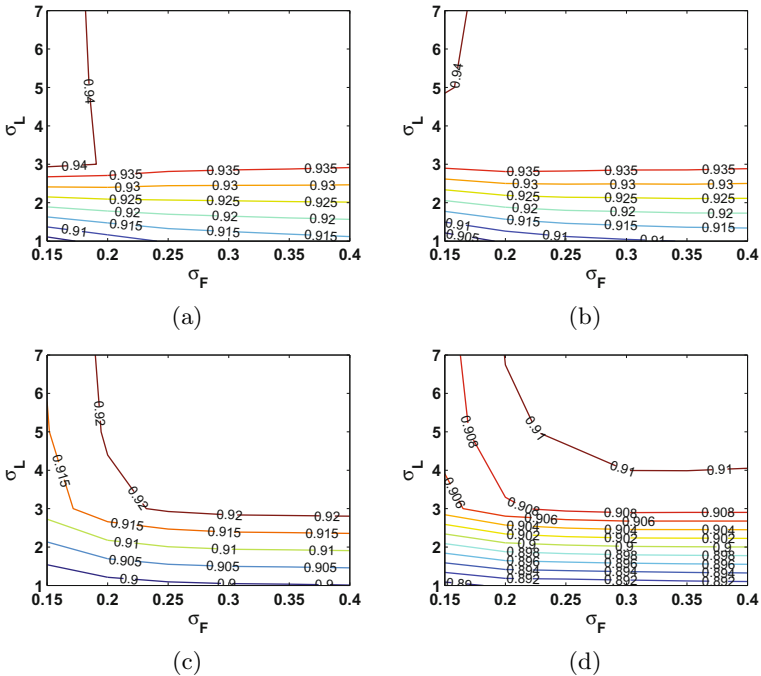


Fig. 3. The sensitivity of the segmentation results to the parameters, x -axis and y -axis represents different values of σ_F and σ_L , the colored lines show isolines of Jaccard similarity, and the value of α is set to 0.005 (a), 0.01 (b), 0.03 (c), and 0.05 (d)

computed separately for the testing subjects, the means and standard deviations of the similarity with different methods are shown in Table 1. Both the visual inspection and the quantitative measure indicate that our method can achieve better tumor segmentation.

Finally, we study how the parameters affect the performance of our method. As shown in Fig. 3, the segmentation performance is robust to the trade-off parameter α . The algorithm with α set on a scale of 0.01 is stable and the segmentation performance varies little with the values of σ_L and σ_F within a wide range.

3.2 Evaluation on Real MR Images

The real MR image dataset contains 5 subjects with tumors of different sizes and locations, each subject has three images including T1-weighted, T2-weighted, and fluid attenuation inversion recovery (FLAIR) image. Tumor region for each subject is manually delineated by 2 raters for training and result validation.

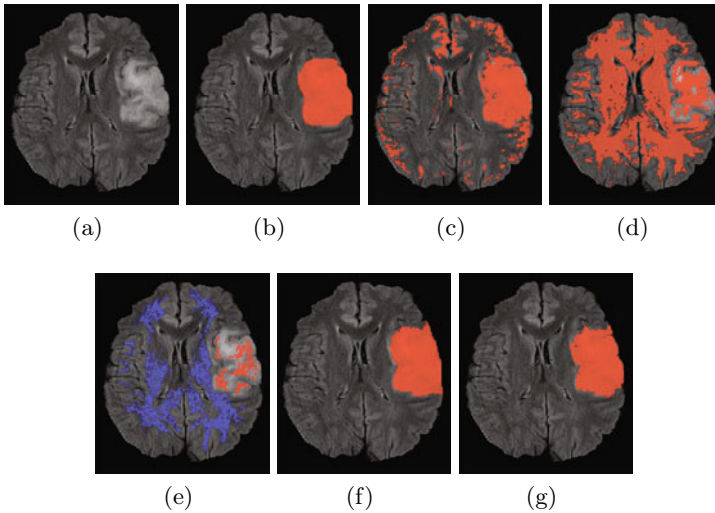


Fig. 4. One slice of a testing real image, manual segmentation, and segmentation results by different methods. (a) the original FLAIR image, (b) one manual segmentation, (c) segmentation obtained by the SVM classification, (d) segmentation using FAST, (e) the initial labeled image for information propagation, (f) segmentation using information propagation method without gradient information used in edge weight measurement, and (g) segmentation of our method.

The whole segmentation process is similar to that applied on the simulate dataset, and the only modification is: the non-brain parts are removed by the BET [23] with manual editing. Due to the large slice thickness of the FLAIR image, $5 \times 5 \times 3$ neighborhood is used for the graph construction.

Fig. 4 shows a representative slice of the real MR images, and its corresponding segmentation results using different methods. The quantitative segmentation

performances estimated by comparing the segmentation results with manual segmentations (rater 1 and 2), including mean Jaccard similarities and standard deviations, are shown in Table 2. The mean Jaccard similarity and standard deviation between the segmentation results of 2 raters is 0.77 and 0.034. From the table, it can be observed that our segmentation method achieved relatively stable and accurate performance.

Table 2. Mean and standard deviation of Jaccard similarity compared with the segmentation of rater 1 and rater 2 on real MR images

Method	Mean (1)	Mean (2)	Std (1)	Std (2)
SVM based segmentation	0.35	0.32	0.151	0.13
FAST	0.17	0.16	0.113	0.114
Traditional information propagation	0.71	0.668	0.004	0.097
Our method	0.76	0.717	0.07	0.029

4 Conclusion

We have presented an information propagation based tumor segmentation method which employs the reliable statistical information from the training data and specific information from the image to be segmented. This method can exploit the local and global consistency of the image specific information, facilitating accurate and reliable tumor segmentation. While the statistical information can provide reasonable initialization for the label information propagation and make it automatic, the image specific information makes up for the insufficient statistical information from the training process and improves the final segmentation performance. The algorithm has been applied to MR image data with tumors of different sizes and locations. The experimental results have demonstrated our method can achieve better tumor segmentation performance, compared with state-of-the-art tumor segmentation methods.

Situations affecting the performance of this method have been encountered in the experiments. For example, the SVM based segmentation will fail when statistical intensity distributions of training and testing data do not match very well. In this case, the proposed method can be adopted as an interactive method whose label initialization is provided by the user input. Furthermore, tumor of different types or grades may affect the performance of the SVM based segmentation method, and subsequently the label propagation process. However, these problems can be alleviated by properly defined image features other than the image intensity only. In this study image intensity information is used as image feature and multi-parametric images are used equivalently. Future work will be devoted to optimally combining multi-parametric MR images and different image modalities.

Acknowledgement. This study was supported in part by the National Science Foundation of China (Grant number: 30970770) and the Hundred Talents Programs, Chinese Academy of Sciences.

References

1. Ahmed, M., Mohamad, D.: Segmentation of brain MR images for tumor extraction by combining Kmeans clustering and Perona-Malik anisotropic diffusion model. *International Journal of Image Processing* 2, 27–34 (2008)
2. Ayed, B., Li, S., Ross, I.: A statistical overlap prior for variational image segmentation. *International Journal of Computer Vision* 85, 115–132 (2009)
3. Black, M.J., Sapiro, G., Marimont, D.H., Heeger, D.: Robust anisotropic diffusion. *IEEE Transactions on Image Process* 7, 421–432 (1998)
4. Chang, C., Lin, C.: Libsvm: a library for support vector machines (2001)
5. Chapelle, O., Weston, J., Scholkopf, B.: Cluster kernels for semi-supervised learning. In: *Advances in Neural Information Processing Systems*, pp. 585–592 (2003)
6. Clark, M., Hall, L., Goldgof, D., Velthuizen, R., Murtagh, F., Silbiger, M.: Automatic tumor segmentation using knowledge-based techniques. *IEEE Transactions on Medical Imaging* 17, 187–201 (1998)
7. Cobzas, D., Birkbeck, N., Schmidt, M., Jagersand, M., Murtha, A.: 3D variational brain tumor segmentation using a high dimensional feature set. In: *IEEE International Conference on Computer Vision*, pp. 1–8 (2007)
8. Dickson, S., Thomas, B.: Using neural networks to automatically detect brain tumors in MR images. *International Journal of Neural Systems* 8, 91–99 (1997)
9. Fletcher-Heath, L., Hall, L., Goldgof, D., Murtagh, F.: Automatic segmentation of non-enhancing brain tumors in magnetic resonance images. *Artificial Intelligence in Medicine* 21, 43–63 (2001)
10. Gering, D.T., Grimson, W.E.L., Kikinis, R.: Recognizing deviations from normalcy for brain tumor segmentation. In: Dohi, T., Kikinis, R. (eds.) *MICCAI 2002*. LNCS, vol. 2488, p. 388. Springer, Heidelberg (2002)
11. Grady, L.: Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28, 1768–1783 (2006)
12. Ho, S., Bullitt, E., Gerig, G.: Level-set evolution with region competition: Automatic 3-D segmentation of brain tumors. In: *International Conference on Pattern Recognition*, pp. 532–535 (2002)
13. Kaus, M., Warfield, S., Nabavi, A., Black, P., Jolesz, F., Kikinis, R.: Automated segmentation of MR images of brain tumors. *Radiology* 218, 586–591 (2001)
14. Lao, Z., Shen, D., Liu, D., Jawad, E., Melhem, E., Launer, L., Bryan, R., Davatzikos, C.: Computer-assisted segmentation of white matter lesions in 3D MR images using support vector machine. *Academic Radiology* 15, 300–313 (2008)
15. Liu, J., Udupa, J., Odhner, D., Hackney, D., Moonis, G.: A system for brain tumor volume estimation via MR imaging and fuzzy connectedness. *Computerized Medical Imaging and Graphics* 29, 21–34 (2005)
16. Moon, N., Bullitt, E., Leemput, K., Gerig, G.: Model-based brain and tumor segmentation. In: *International Conference on Pattern Recognition*, pp. 528–531 (2002)
17. Nie, J., Xue, Z., Liu, T., Young, G., Setayesh, K., Guo, L., Wong, S.: Automated brain tumor segmentation using spatial accuracy-weighted hidden Markov random field. *Computerized Medical Imaging and Graphics* 33, 431–441 (2009)
18. Prastawa, M., Bullitt, E., Ho, S., Gering, G.: A brain tumor segmentation framework based on outlier detection. *Medical Image Analysis* 8, 275–283 (2004)
19. Prastawa, M., Bullitt, E., Gerig, G.: Synthetic ground truth for validation of brain tumor MRI segmentation. In: Duncan, J.S., Gerig, G. (eds.) *MICCAI 2005*. LNCS, vol. 3749, pp. 26–33. Springer, Heidelberg (2005)

20. Rousseeuw, P.J., Driessen, K.V.: A fast algorithm for the minimum covariance determinant estimator. *Technometrics* 41, 212–223 (1999)
21. Sapiro, G., Ringach, D.: Anisotropic diffusion of multivalued images with applications to color filtering. *IEEE Transactions on Image Processing* 5, 1582–1586 (1996)
22. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 888–905 (2000)
23. Smith, S.: Fast robust automated brain extraction. *Human Brain Mapping* 17, 143–155 (2002)
24. Taheri, S., Ong, S., Chong, V.: Level-set segmentation of brain tumors using a threshold-based speed function. *Image and Vision Computing* 28, 26–37 (2010)
25. Wu, Z., Leahy, R.: An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15, 1101–1113 (1993)
26. Zhang, Y., Brady, M., Smith, S.: Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm. *IEEE Transactions on Medical Imaging* 20, 45–57 (2001)
27. Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schlkopf, B.: Learning with local and global consistency. In: *Advances in Neural Information Processing Systems*, pp. 321–328 (2004)
28. Zhu, Y., Yan, H.: Computerized tumour boundary detection using a Hopfield neural network. In: *IEEE International Conference on Neural Networks*, pp. 2467–2472 (1995)

Face Recognition with Decision Tree-Based Local Binary Patterns

Daniel Maturana, Domingo Mery, and Álvaro Soto

Department of Computer Science, Pontificia Universidad Católica de Chile

Abstract. Many state-of-the-art face recognition algorithms use image descriptors based on features known as Local Binary Patterns (LBPs). While many variations of LBP exist, so far none of them can automatically adapt to the training data. We introduce and analyze a novel generalization of LBP that learns the most discriminative LBP-like features for each facial region in a supervised manner. Since the proposed method is based on Decision Trees, we call it Decision Tree Local Binary Patterns or DT-LBPs. Tests on standard face recognition datasets show the superiority of DT-LBP with respect of several state-of-the-art feature descriptors regularly used in face recognition applications.

1 Introduction

While face recognition algorithms commonly assume that face images are well aligned and have a similar pose, in many practical applications it is impossible to meet these conditions. Therefore extending face recognition to less constrained face images has become an active area of research.

To this end, face recognition algorithms based on properties of small regions of face images – often known as local appearance descriptors or simply local descriptors – have shown excellent performance on standard face recognition datasets. Examples include the use Gabor features [30], SURF [4,8], SIFT [14,5], HOG [7,3], and histograms of Local Binary Patterns (LBPs) [17,2]. A comparison of various local descriptor-based face recognition algorithms may be found in Ruiz del Solar et al [20].

Among the different local descriptors in the literature, histograms of LBPs have become popular for face recognition tasks due to their simplicity, computational efficiency, and robustness to changes in illumination. The success of LBPs has inspired several variations. These include local ternary patterns [23], elongated local binary patterns [12], multi scale LBPs [13], patch based LBPs [24], center symmetric LBPs [10] and LBPs on Gabor magnitude images [29,26], to cite a few. However, these are specified a priori without any input from the data itself, except in the form of cross-validation to set parameters.

In this paper, our main contribution is to propose a new method that explicitly learns discriminative descriptors from the training data. This method is based on a connection between LBPs and decision trees. As a testing scenario, we consider the traditional task of *closed set face identification*. Under this task, we are given

a gallery of identified face images, such that, for any unidentified probe image, the goal is to return one of the identities from the gallery.

This paper is organized as follows. Section 2 presents general background information about the operation of traditional LBPs and also about the pipeline used by our approach to achieve face recognition. Section 3 presents the main details of our approach. Section 4 discusses relevant previous work. Section 5 shows the main experiments and results of applying our approach to two standard benchmark datasets. Finally, Section 6 presents the main conclusions of this work.

2 Background Information

2.1 Local Binary Patterns

Local binary patterns were introduced by Ojala et al [17] as a fine scale texture descriptor. In its simplest form, an LBP description of a pixel is created by thresholding the values of a 3×3 neighborhood with respect its central pixel and interpreting the result as a binary number.

In a more general setting, a LBP operator assigns a decimal number to a pair (c, \mathbf{n}) ,

$$b = \sum_{i=1}^S 2^{i-1} I(c, n_i)$$

where c represents a center pixels, $\mathbf{n} = (n_1, \dots, n_S)$ corresponds to a set of pixels sampled from the neighborhood of c according to a given pattern, and

$$I(c, n_i) = \begin{cases} 1 & \text{if } c < n_i \\ 0 & \text{otherwise} \end{cases}$$

This can be seen as assigning a 0 to each neighbor pixel in \mathbf{n} that is larger than the center pixel c , a 1 to each neighbor smaller than c , and interpreting the result as a number in base 2. In this way, for the case of a neighborhood of S pixels, there are 2^S possible LBP values.

2.2 Face Recognition Pipeline

Our face recognition pipeline is similar to the one proposed in [2], but we incorporate a more sophisticated illumination normalization step [23]. Figure 1 summarizes its operation, given by the following main steps:

1. Crop the face region and align the face by mapping the eyes to a canonical location with a similarity transform.
2. Normalize illumination with Tan and Triggs' [23] Difference of Gaussians filter.
3. Partition the face image in a grid with equally sized cells, the size of which is a parameter.

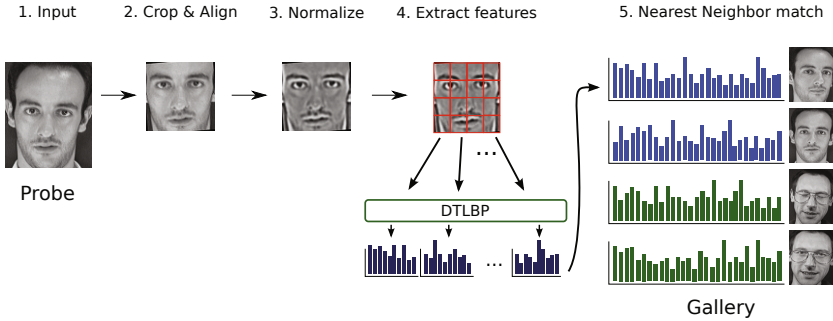


Fig. 1. Our face recognition pipeline

4. For each grid cell, apply a feature extraction operator (such as LBPs) to each pixel in the grid cell. Afterward, create a histogram of the feature values and concatenate these histograms into a single vector, usually known as “spatial histogram”.
5. Classify a probe face with the identity of the nearest neighbor in the gallery, where the nearest neighbor distance is calculated with the (possibly weighted) L_1 distance between the histograms of the corresponding face images.

3 Our Approach: Decision Tree Local Binary Patterns

The simple observation behind DT-LBP is that the operation of a LBP over a given neighborhood is equivalent to the application of a fixed binary decision tree. In effect, the aforementioned histograms of LBPs may be seen as quantizing each pair (c, \mathbf{n}) with a specially constructed binary decision tree, where each possible branch of the tree encodes a particular LBP. The tree has S levels, where all the nodes at a generic level l compare the center pixel c with a given neighbor $n_l \in \mathbf{n}$. In this way, at each level $l - 1$, the decision is such that, if $c < n_l$ the vector is assigned to the left node; otherwise, it is assigned to the right node. Since the tree is complete, at level 0 we have 2^S leaf nodes. Each of these nodes corresponds to one of the 2^S possible LBPs. In fact, seen as a binary number, each LBP encodes the path taken by (c, \mathbf{n}) through the tree; for example, in a LBP with $S = 8$, 11111101 corresponds to a (c, \mathbf{n}) pair which has taken the left path at level $l = 1$ and taken the right path at all other levels.

The previous equivalence suggests the possibility of using standard decision tree induction algorithms in place of a fixed tree to learn discriminative LBP-like descriptors from training data. We call this approach Decision Tree Local Binary Patterns or DT-LBP. As a major advantage, by using training data to learn the structure of the tree, DT-LBP can effectively build an adaptive tree, whose main branches are specially tuned to encode discriminative patterns for the relevant target classes. Furthermore, the existence of efficient algorithms to train a decision tree allows DT-LBP to explore larger neighborhoods, such that, at the end of the process the resulting structure of the tree and corresponding pixel comparisons at each node provide more discriminative *spatial histograms*.

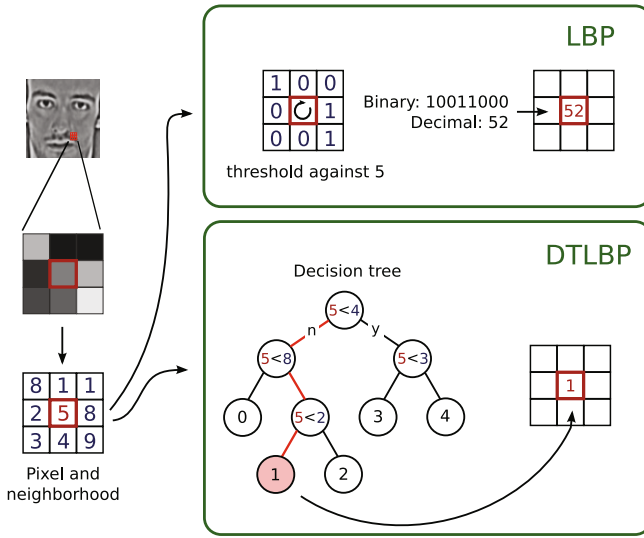


Fig. 2. The LBP operator versus the DT-LBP operator

Figure 2 compares the operation of regular LBPs with respect to DT-LBPs. After a decision tree is trained, DT-LBP assigns to each leaf node a code given by the path or branch that leads to that node in the tree. In this way, for any input pixel c and the corresponding neighborhood \mathbf{n} used to build the tree, the pair (c, \mathbf{n}) moves down the tree according to the $c < n_i$ comparisons. Once it reaches a leaf node, the respective code is assigned to the center pixel c (code number 1 in Figure 2). As with ordinary LBPs, the DT-LBPs obtained for a given image can be used for classification by building histograms. In summary the proposed approach has the following advantages:

- We can obtain adaptive and discriminative LBPs by leveraging well known decision tree construction algorithms (e.g. [19]), as well as more recent randomized tree construction algorithms that have been shown to be very effective in computer vision applications (e.g. [16]).
- Since we expect different patterns to be discriminative in different face image regions, we can learn a different tree for each region.
- Instead of neighborhood of eight or sixteen pixels as in regular LBPs, we can use a much larger neighborhood and let the tree construction algorithm decide which neighbors are more relevant.
- Apart from the feature extraction step, DT-LBP can be used with no modification in any of the many applications where LBP is currently applied.

3.1 Tree Learning Details

To maximize the adaptivity of our algorithm we learn a tree for each grid cell. The trees are recursively built top-down with a simple algorithm based

on Quinlan’s classic ID3 method [19]. The algorithm takes as input a “dataset” $\mathcal{X} = \{(c_i, \mathbf{n}_i, y_i)\}_{i=1}^N$, a set of tuples where c_i is the value of the center pixel, $\mathbf{n}_i = (n_{i1}, \dots, n_{is})$ is the vector of values of c_i ’s neighbors, and y_i is the label of the image from which c_i is taken. These values are taken from the pixels in each grid cell of the images in the training data. The following pseudocode summarizes the algorithm:

```

build_tree( $\mathcal{X}$ )  $\equiv$ 
{Recursively build DT-LBP tree}
if terminate then
  return LeafNode
else
   $m \leftarrow$  choose_split( $\mathcal{X}$ )
  left  $\leftarrow$  build_tree( $\{(c_i, \mathbf{n}_i, y_i) \in \mathcal{X} \mid c_i \geq n_{im}\}$ )
  right  $\leftarrow$  build_tree( $\{(c_i, \mathbf{n}_i, y_i) \in \mathcal{X} \mid c_i < n_{im}\}$ )
  return SplitNode( $m, \text{left}, \text{right}$ )
end if

choose_split( $\mathcal{X}$ )  $\equiv$ 
{Choose most informative pixel comparison}
for  $d = 1$  to  $S$  do
   $\mathcal{X}_L \leftarrow \{(c_i, \mathbf{n}_i, y_i) \in \mathcal{X} \mid c_i \geq n_{id}\}$ 
   $\mathcal{X}_R \leftarrow \{(c_i, \mathbf{n}_i, y_i) \in \mathcal{X} \mid c_i < n_{id}\}$ 
   $\Delta H_d \leftarrow H(\mathcal{X}) - \frac{|\mathcal{X}_L|}{|\mathcal{X}|} H(\mathcal{X}_L) - \frac{|\mathcal{X}_R|}{|\mathcal{X}|} H(\mathcal{X}_R)$ 
end for
return  $\arg \max_d \Delta H_d$ 

```

where $H(\mathcal{X})$ is the class entropy impurity of \mathcal{X} , i.e. $H(\mathcal{X}) = -\sum_{\omega} p(\omega) \lg p(\omega)$, $p(\omega)$ being the fraction of tuples in \mathcal{X} with class label $y_i = \omega$. **terminate** yields true if a maximum depth is reached, $|\mathcal{X}|$ is smaller than a size threshold, or there are no informative pixel comparisons available¹. The size threshold for $|\mathcal{X}|$ is fixed as 10, and the maximum depth is a parameter.

We define the neighborhood \mathbf{n} used by DT-LBP somewhat differently than LBPs. We use a square neighborhood centered around c , and instead of samples taken along a circle, as in regular LBPs, we consider all pixels inside the square as part of the neighborhood (fig. (3)). All the pixels within this square are considered as potential split candidates. The idea is to let the tree construction algorithm find the most discriminative pixel comparisons.

The main parameters of this algorithm are the size of the neighborhood \mathbf{n} to explore, and the maximum depth of the trees. As shown in Figure 3, the first parameter is determined by a radius r . The second parameter, tree depth, determines the size of the resulting histograms. Smaller histograms are desirable for space and time efficiency, but as we will show in our experiments, there is a trade-off in accuracy with respect to larger histograms.

¹ Once a pixel comparison is chosen for a tree node, it provides no information for the descendants of the node.

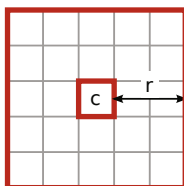


Fig. 3. Pixel neighborhood used in DT-LBP. The inner square is the center pixel c , and the neighborhood corresponds to all the pixels enclosed in the larger square. The size of the neighborhood is determined by the radius r .

Using trees opens up various possibilities. We have explored some extensions to the basic idea, such as using a forest of randomized trees (as in [22] and [16]), trees splitting based on a linear combinations of the values of the neighborhood (i.e. nodes split on $\mathbf{n}^T \mathbf{w} < c$, similarly to [6]), or using ternary trees where a middle branch corresponds to pairs for which $|c - n_i| < \epsilon$ for a small ϵ . This last approach can be considered as the tree-based version of the local ternary patterns described in [23]. So far, we have found that a single tree built with an ID3-style algorithm is the best performing solution.

4 Related Work

Our algorithm can be seen as a way to quantize (c, \mathbf{n}) pairs using a codebook where each code corresponds to a leaf node. This links our algorithm to various other works in vision that use codebooks of image features to describe the images.

Forests of randomized trees have become a popular option to construct codebooks for computer vision tasks. Moosmann et al [16] use Extremely Randomized Clustering forests to create codebooks of SIFT descriptors [14]. Shotton et al. [22] use random forests to create codebooks for use as features in image segmentation. While the use of trees in these works is similar to ours, they use the results of the quantization in a very different way; the features are given to classifiers such as SVMs, which are not suitable for use in our problem. Furthermore, we have found that for our problem single trees are more effective than random forests.

Wright and Hua [25] use unsupervised random forests to quantize SIFT-like descriptors for face recognition. The main difference with our algorithm, besides the use of forests versus single trees, is that we do not quantize complex descriptors extracted from the image but work directly on the image itself. In addition, their trees use decision planes as opposed to simple pixel comparisons, which are faster.

There are various recent works using K-Means to construct codebooks to be used for face recognition in a framework similar to ours. Ahonen et al [1] proposed to view the difference $c - n_i$ of each neighbor pixel n_i with the center as the approximate response of a derivative filter centered on c . Under this view, the LBP operator is a coarse way to quantize the joint responses of various filters

(one for each neighbor n_i). Likewise, DT-LBP is also a quantizer of these joint responses, but it is built adaptively. Ahonen tested the K-Means algorithm as an adaptive quantizer, but did not find it to be clearly superior to LBPs for a face recognition task. Meng et al [15] use K-Means to directly quantize patches from the grayscale image. Xie et al [26,27], as well as [11] use it to quantize patches from images convolved with Gabor wavelets at various scales and orientations. These algorithms are the closest in spirit to our work, since they are partly inspired by LBPs. These algorithms differ from ours in the algorithm used to construct the codebook. They use K-Means, which has the drawback of not being supervised and thus unable to take advantage of labeled data. In addition, trees are more efficient; the time required to quantize a patch with K-Means increases linearly with the size of the codebook, whereas with trees it increases logarithmically. Finally, unlike ours, various of the above algorithms use a bank of Gabor wavelet filters. The convolution of each image with the filter bank adds substantial processing time.

5 Experiments

We perform experiments on the FERET [18] and the CAS-PEAL-R1 [9] benchmark databases. First, we examine the effects of the two main parameters of DT-LBP: the radius r and the maximum tree depth d . In this case, we measure the accuracy of the algorithm on a subset of FERET. Afterward, we report the accuracy of our algorithm on various standard subsets of FERET and CAS-PEAL-R1 with a selected set of parameters.

In all images we partition the image into an 7×7 grid. We tested this partition after evaluating partitions of 6×6 , 7×7 , 8×8 and 9×9 in the AT&T/ORL face dataset [21] with an exhaustive grid search over various maximum tree depth and radii combinations. The mean accuracy over all these combinations, with 5 training images and 5 testing images, was .954, .972, .933 and .931 respectively. While in general we have found this partition to provide good results, it is likely that adjusting the grid size to each database may yield better results.

For each experiment we show our results along with results from similar works in the recent literature: the original LBP algorithm from Ahonen [2]; the Local Gabor Binary Pattern (LGBP) algorithm, which applies LBP to Gabor-filtered images; the Local Visual Primitive (LVP) algorithm of Meng et al [15], which uses K-Means to quantize grayscale patches; the Local Gabor Textons (LGT) algorithm and the Learned Local Gabor Pattern (LLGP) algorithms, which use K-Means to quantize Gabor filtered-images; and the Histogram of Gabor Phase Patterns (HGPP) algorithms, which quantizes Gabor filtered images into histograms that encode not only the magnitude, but also the phase information from the image.

The results are not strictly comparable, since there may be differences in preprocessing and other details, but they provide a meaningful reference. It is worth noting that for each of the algorithm we only show non-weighted variants, since our algorithm does not currently incorporate weights for different facial regions.

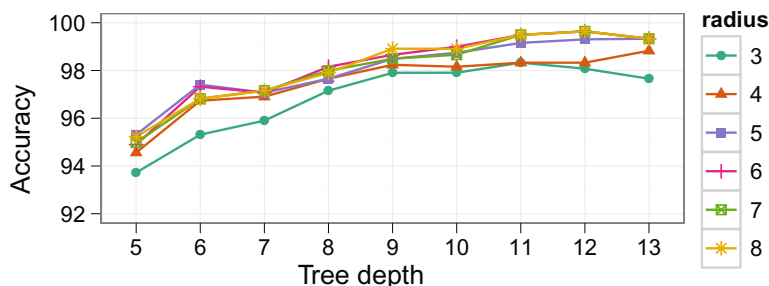


Fig. 4. Effect on accuracy of radius and maximum tree depth in FERET fb

5.1 Effect of Tree Depth and Neighborhood Size

Figure 4 shows the accuracy obtained on FERET *fb* with various combinations of neighborhood sizes and depths. While neighborhood sizes of $r = 1$ and $r = 2$ were also tested, as expected these perform poorly with large trees and are not shown.

We see that larger trees tend to boost performance, however, for some radii there is a point where larger trees decrease accuracy. This suggests that overfitting may be occurring for these radii sizes. We also see that while larger radii tend to perform better, all radii larger than 6 perform similarly. Therefore we set the radius to 7 pixels in the following two experiments.

5.2 Results on FERET

For FERET, we use *fa* as gallery and *fb*, *fc*, *dup1* and *dup2* as probe sets. For training, we use the FERET standard training set of 762 images from the training CD provided by the CSU Face Identification Evaluation System package.

We can see that our algorithm relies on the Tan-Triggs normalization step to obtain competitive results on the probe sets with heavy illumination variation. Note that the results we show without normalization use no normalization at all. When combined with the normalization step, our algorithm obtains the best results on all the probe sets. We argue that the Difference-of-Gaussian filter in the Tan-Triggs normalization plays a similar role to the Gabor filters in the Gabor-based algorithms, but is much more efficient computationally.

5.3 Results on CAS-PEAL-R1

Again, our algorithm is affected by intense illumination variation when used without normalization. With normalization our algorithm performs better in the Expression probe set and comparably with HGPP in the Accessory probe set. On the lighting dataset, the overall performance of all the algorithms is rather poor. In this case, the best results are given by LGBP, HGPP and LLGP. All these algorithms use features based on Gabor wavelets, which suggests that Gabor features provide more robustness against the extreme lighting variations in this dataset than the DoG filter.

Table 1. Accuracy on FERET probe sets. DT-LBP $_d^r$ corresponds to a tree of maximum depth d and radius r . *TT* indicates Tan-Triggs DoG normalization. Accuracies for algorithms other than DT-LBP come from the cited papers.

Method	fb	fc	dup1	dup2
LBP [2]	0.93	0.51	0.61	0.50
LGBP [29]	0.94	0.97	0.68	0.53
LVP [27]	0.97	0.70	0.66	0.50
LGT [11]	0.97	0.90	0.71	0.67
HGPP [28]	0.98	0.99	0.78	0.76
LLGP [27]	0.97	0.97	0.75	0.71
DT-LBP $_8^7$, no TT	0.98	0.44	0.63	0.42
DT-LBP $_{10}^7$, no TT	0.98	0.55	0.65	0.47
DT-LBP $_{12}^7$, no TT	0.99	0.63	0.67	0.48
DT-LBP $_8^7$	0.98	0.99	0.79	0.78
DT-LBP $_{10}^7$	0.99	0.99	0.83	0.78
DT-LBP $_{12}^7$	0.99	1.00	0.84	0.79
DT-LBP $_{13}^7$	0.99	1.00	0.84	0.80

Table 2. Accuracy on CAS-PEAL-R1 probe sets. DT-LBP $_d^r$ corresponds to a tree of maximum depth d and radius r . Accuracies for algorithms other than DT-LBP come from the cited papers. *TT* indicates Tan-Triggs DoG normalization.

Method	Expression	Accessory	Lighting
LGBP [29]	0.95	0.87	0.51
LVP [15]	0.96	0.86	0.29
HGPP [28]	0.96	0.92	0.62
LLGP [27]	0.96	0.90	0.52
DT-LBP $_8^7$, no TT	0.96	0.80	0.20
DT-LBP $_{10}^7$, no TT	0.99	0.87	0.23
DT-LBP $_{12}^7$, no TT	0.99	0.88	0.25
DT-LBP $_8^7$	0.95	0.89	0.36
DT-LBP $_{10}^7$	0.98	0.91	0.39
DT-LBP $_{12}^7$	0.98	0.92	0.40
DT-LBP $_{13}^7$	0.98	0.92	0.41

5.4 Discussion

The results show that DT-LBPs are highly discriminative features. Their discriminativity increases as the trees grow, but this has an exponential impact in the computational time and storage cost of using these features. For example, a tree of maximum depth 8 corresponds to a maximum of 256 histogram bins, while a tree with maximum depth 14 corresponds to a maximum of 16384 bins. Since we use $7 \times 7 = 49$ grid cells, the total number of histogram bins in each spatial histogram is around 802,816 bins. In practice, we find that our C++

implementation is fast enough for many applications – converting an image to a DT-LBP spatial histogram and finding its nearest neighbor in a gallery with more than a thousand images takes a couple of seconds. However, the cost in terms of memory and storage becomes an obstacle to the use of larger trees. For example, a gallery of 1196 subjects with 49 grid cells and trees of maximum depth 14 takes about 3.5 GB of storage when stored naively. However, the resulting dataset is very sparse, which can be taken advantage of to compress it. A straightforward solution is not to use in our coding all the branches of the resulting trees but only the most popular ones. This is a similar simplification to the one used by traditional LBPs through the so-called *uniform patterns*.

6 Conclusions and Future Work

We have proposed a novel method that uses training data to create discriminative LBP-like descriptors by using decision trees. The algorithm obtains encouraging results on standard databases, and presents better results than several state-of-the-art alternative solutions.

As future work, our current implementation does not assign different weights to different face regions. Incorporating weights has been shown to be an effective strategy in various similar works, such as [1] and [27], so we plan to explore the addition of weights. Furthermore, we are currently working on reducing the size of the resulting histograms while maintaining or improving accuracy. To achieve this we are exploring different methods to learn the decision trees and other data structures to represent adaptable LBP-like descriptors. Finally, seeing the good performance of algorithms that use features based on Gabor wavelets (such as [27] and [28]) we are incorporating these type of features into our algorithm.

Acknowledgements. This work was partially funded by FONDECYT grant 1095140 and LACCIR Virtual Institute grant No. R1208LAC005 (<http://www.laccir.org>). The authors thank Pontificia Universidad Católica de Chile for the VRI Grant and Pontificia Universidad Católica's School of Engineering for the FIA Grant. Portions of the research in this paper use the FERET database of facial images collected under the FERET program. The research in this paper uses the CAS-PEAL-R1 face database collected under the sponsorship of the Chinese National Hi-Tech Program and ISVISION Tech. Co. Ltd. Finally, we thank the anonymous reviewers for their helpful comments.

References

1. Ahonen, T., Pietikäinen, M.: Image description using joint distribution of filter bank responses. *Pattern Recognition Letters* 30, 368–376 (2009)
2. Ahonen, T., Hadid, A., Pietikäinen, M.: Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28, 2037–2041 (2006)
3. Albiol, A., Monzo, D., Martin, A., Sastre, J., Albiol, A.: Face recognition using HOG-EBGM. *Pattern Recognition Letters* 29, 1537–1543 (2008)

4. Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding* 110, 346–359 (2008)
5. Bicego, M., Lagorio, A., Grosso, E., Tistarelli, M.: On the use of SIFT features for face authentication. In: *CVPR* (2006)
6. Bosch, A., Zisserman, A., Munoz, X.: Image classification using random forests and ferns. In: *ICCV* (2007)
7. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: *CVPR* (2005)
8. Dreuw, P., Steingrube, P., Hanselmann, H., Ney, H.: SURF-face: Face recognition under viewpoint consistency constraints. In: *BMVC* (2009)
9. Gao, W., Cao, B., Shan, S., Chen, X., Zhou, D., Zhang, X., Zhao, D.: The CAS-PEAL large-scale chinese face database and baseline evaluations. *IEEE Transactions on System Man, and Cybernetics (Part A)* 38, 149–161 (2008)
10. Heikkilä, M., Pietikäinen, M., Schmid, C.: Description of interest regions with local binary patterns. *Pattern Recognition* 42, 425–436 (2009)
11. Lei, Z., Li, S.Z., Chu, R., Zhu, X.: Face recognition with local gabor textons. In: Lee, S.-W., Li, S.Z. (eds.) *ICB 2007*. LNCS, vol. 4642, pp. 49–57. Springer, Heidelberg (2007)
12. Liao, S., Chung, A.C.S.: Face recognition by using elongated local binary patterns with average maximum distance gradient magnitude. In: Yagi, Y., Kang, S.B., Kweon, I.S., Zha, H. (eds.) *ACCV 2007, Part II*. LNCS, vol. 4844, pp. 672–679. Springer, Heidelberg (2007)
13. Liao, S., Zhu, X., Lei, Z., Zhang, L., Li, S.Z.: Learning multi-scale block local binary patterns for face recognition. In: Lee, S.-W., Li, S.Z. (eds.) *ICB 2007*. LNCS, vol. 4642, pp. 828–837. Springer, Heidelberg (2007)
14. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60, 91–110 (2004)
15. Meng, X., Shan, S., Chen, X., Gao, W.: Local Visual Primitives (LVP) for face modelling and recognition. In: *ICPR* (2006)
16. Moosmann, F., Nowak, E., Jurie, F.: Randomized clustering forests for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30, 1632–1646 (2008)
17. Ojala, T., Pietikäinen, M., Harwood, D.: A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition* 29, 51–59 (1996)
18. Phillips, P.J., Moon, H., Rizvi, S.A., Rauss, P.J.: The FERET evaluation methodology for Face-Recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 1090–1104 (2000)
19. Quinlan, J.R.: Induction of decision trees. *Machine Learning* 1, 81–106 (1986)
20. Ruiz-del-Solar, J., Verschae, R., Correa, M.: Recognition of faces in unconstrained environments: A comparative study. *EURASIP Journal on Advances in Signal Processing* 2009, 1–20 (2009)
21. Samaria, F., Harter, A.: Parameterisation of a stochastic model for human face identification. In: *Proc. Second IEEE Workshop on Applications of Computer Vision*, pp. 138–142 (1994)
22. Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image categorization and segmentation. In: *CVPR* (2008)
23. Tan, X., Triggs, B.: Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Transactions on Image Processing* 19, 1635–1650 (2010)

24. Wolf, L., Hassner, T., Taigman, Y.: Descriptor based methods in the wild. In: Real-Life Images Workshop at ECCV (2008)
25. Wright, J., Hua, G.: Implicit elastic matching with random projections for pose-variant face recognition. In: CVPR, pp. 1502–1509 (2009)
26. Xie, S., Shan, S., Chen, X., Gao, W.: V-LGBP: Volume based Local Gabor Binary Patterns for face representation and recognition. In: ICPR (2008)
27. Xie, S., Shan, S., Chen, X., Meng, X., Gao, W.: Learned local Gabor patterns for face representation and recognition. *Signal Processing* 89, 2333–2344 (2009)
28. Zhang, B., Shan, S., Chen, X., Gao, W.: Histogram of Gabor phase patterns (HGPP): A novel object representation approach for face recognition. *IEEE Transactions on Image Processing* 16, 57–68 (2007)
29. Zhang, W., Shan, S., Gao, W., Chen, X., Zhang, H.: Local Gabor Binary Pattern Histogram Sequence (LGBPHS): A novel non-statistical model for face representation and recognition. In: ICCV (2005)
30. Zou, J., Ji, Q., Nagy, G.: A comparative study of local matching approach for face recognition. *IEEE Transactions on Image Processing* 16, 2617–2628 (2007)

Occlusion Handling with ℓ_1 -Regularized Sparse Reconstruction

Wei Li¹, Bing Li¹, Xiaoqin Zhang², Weiming Hu¹,
Hanzi Wang³, and Guan Luo¹

¹ National Lab of Pattern Recognition, Institute of Automation, CAS, Beijing, China
{weili,wmhu,gluo}@nlpr.ia.ac.cn

² College of Mathematics & Information Science, Wenzhou University,
Zhejiang, China
xqzhang@wzu.edu.cn

³ Cognitive Science Department, School of Information Science and Technology,
Xiamen University

³ Fujian Key Lab of the Brain-Like Intelligent Systems, Xiamen, China
Hanzi.Wang@ieee.org

Abstract. Tracking multi-object under occlusion is a challenging task. When occlusion happens, only the visible part of occluded object can provide reliable information for the matching. In conventional algorithms, the deducing of the occlusion relationship is needed to derive the visible part. However deducing the occlusion relationship is difficult. The inter-determined effect between the occlusion relationship and the tracking results will degenerate the tracking performance, and even lead to the tracking failure. In this paper, we propose a novel framework to track multi-object with occlusion handling according to sparse reconstruction. The matching with ℓ_1 -regularized sparse reconstruction can automatically focus on the visible part of the occluded object, and thus exclude the need of deducing the occlusion relationship. The tracking is simplified into a joint Bayesian inference problem. We compare our algorithm with the state-of-the-art algorithms. The experimental results show the superiority of our algorithm over other competing algorithms.

1 Introduction

Object tracking is a challenging task in vision systems. It has received significant attention due to its crucial values in many applications such as surveillance, vision-based control, human-computer interfaces, intelligent transportation, and augmented reality.

In recent years, many algorithms have been proposed: for example, template matching [1], mean shift [2], condensation [3], appearance models [4] and so on. These algorithms have achieved great success in tracking field from different perspective. However, it is still a challenging task to design a robust tracking algorithm to track multiple objects under occlusion. This is because during occlusion, only portions of each occluded object are visible and the correspondences between objects and their features become ambiguous.

The usual way to tackle occlusion problem is to model the occlusion relationship between different objects explicitly. Different techniques have been proposed to deduce the occlusion relationship. In [5], each object is modeled as a layer to utilize depth information, and a variable is employed to describe the affiliation of each pixel to the different layers. Elgammal and Davis [6] segment people under occlusion by incorporating the occlusion relationship of different layers into a finely defined likelihood. Wu et al. [7] apply the Bayesian network to track two faces through occlusion in which an extra hidden process for the occlusion representation is introduced. Sudderth et al. [8] propose a looselimb model consisting of a set of connected geometric primitives, and the inference is conducted by using nonparametric belief propagation. In the methods mentioned above, a complex occlusion reasoning framework is required for state inference during occlusion. However it is very difficult to deduce the occlusion relationship. And moreover, if the occlusion reasoning is wrong, it will lead to the failure in tracking.

In order to overcome the drawbacks induced by modeling the occlusion relationship explicitly, many researchers begin to adopt certain rules inspired from other areas to implicitly handle occlusion. MacCormick and Blake [9] develop a probabilistic exclusion principle based data association filter to solve the occlusion problem in multiple object tracking, but it is only applied for two objects. In [10], the spatio-temporal context of each object is used to maintain the correct identity of the object during the occlusion process. Yang et al. [11] propose a game-theoretic multiple target tracking algorithm. Tracking is analogue to find the Nash Equilibrium of a game. The algorithm proposed in this paper falls into this category but does not need to subtly design certain rules.

In this paper, we propose an effective mechanism for multi-object tracking with occlusion handling using sparse reconstruction. The location and the size of the occluded part are robustly explored with ℓ_1 -regularized sparse reconstruction. In essence, the sparse reconstruction can automatically focus on the visible part of the object in matching the warped image with the templates. As a result, there is no need to deduce the occlusion relationship for the matching. Thus, the occlusion state can be eliminated from the tracking inference process. The multi-object tracking with occlusion is reduced to a simple joint state Bayesian inference problem. During the occlusion process, a cross iteration technique is proposed to greatly improve the efficiency. Also different tracking strategies are employed for both mild and severe occlusion. After the objects are localized, the templates of the objects are updated accordingly.

The rest of this paper is organized as follows. In section 2, the introduction of sparse reconstruction is firstly presented. Then the tracking algorithm with occlusion handling is detailed. The experimental results to validate our method are will be presented in Section 3 which follows the conclusion in Section 4.

2 Proposed Algorithm

In this section, we first give a brief review of the sparse representation. Then an observation model based on the sparse reconstruction is introduced. Finally, a multi-object tracking algorithm with occlusion handling is presented.

2.1 Sparse Reconstruction Based Appearance Model

In this paper, we assume the manifold of the same object in different frames lies in a linear subspace. This is reasonable because the variations of the appearance mode are usually reflected on a special low-dimensional subspace. It means that any new sample of the same object with some variations can be approximately spanned by a set of templates. Let $T_{i=1}^n$ represent n object templates selected before tracking, a new image m can be approximated by a linear combination of these templates:

$$m = \omega_1 T_1 + \omega_2 T_2 + \dots + \omega_n T_n + \varepsilon \tag{1}$$

where $\omega = (\omega_1, \omega_2, \dots, \omega_n)^T$ is the coefficient vector, and ε is a noise term.

If the object is occluded by another one, the value of ε is considered to be the discrepancy between the occluded part and the according templates. In the occluded part, the discrepancy is much larger than the un-occluded part. Thus the distribution of ε represents the locations of the occluded pixels. Since the location and the size of the occluded part can differ for different tracking images and are unknown to the computer, a set of trivial templates [12] $I = (I_1, I_2, \dots, I_d)$ are defined to explicitly code the occluded pixels, where d is number of the trivial templates and it equals to the dimension of the template after spanning into a 1D vector. Each trivial template I_i is a vector with only one nonzero entity in the position i . The detailed composition of the trivial templates is shown in Fig.1 . The whole trivial template is represented with a blank template except for the nonzero entity. Then the discrepancy ε can be sparsely coded with the combination of these trivial templates:

$$\varepsilon = [I_1, I_2, \dots, I_d][e_1, e_2, \dots, e_d]^T \tag{2}$$

where e_i is the coefficient of the i th trivial template. In order to unify the scale between the trivial templates I and the object templates T , we normalize each template T_i by subtracting the mean and dividing the covariance of the templates.

From the above definition, the nonzero entity of e model which pixels in m are occluded by the other object. The combination of e and I results in the restoration of ε which represents the value of the discrepancy. So the image m

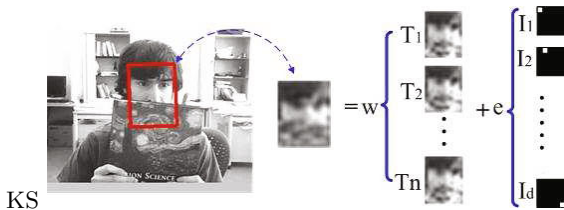


Fig. 1. The object with occlusion is a combination of object templates $T_{i=1}^n$ and trivial templates $I = (I_1, I_2, \dots, I_d)$

is rewritten using the following new form with the occlusion information being included:

$$m = [T \ I] \begin{bmatrix} \omega \\ e \end{bmatrix} \doteq B\rho \quad (3)$$

During the tracking, we intend to choose the most similar template for the matching. So the coefficient ω should be as sparse as possible. Also for the occlusion coefficient e , we intend to constrain it only accounting for the occlusion part. These two requirements can be satisfied by minimizing the ℓ_1 -norm of ρ . At the same time, the tracking result is obtained by minimizing the distance between the candidate region and the templates. This can be achieved through minimizing the ℓ_2 -norm of the residual error $m - B\rho$. According to the above discussion, the minimization problem to obtain the coefficient ρ is defined as follows:

$$\hat{\rho} = \arg \min \|\rho\|_1 + \|m - B\rho\|_2 \quad (4)$$

The above ℓ_1 -norm minimizing problem can be efficiently solved via the linear programming algorithm based on [13].

The distribution of the recovered parameter e in coefficient $\hat{\rho}$ represents the position and the size of the object's occluded part. In the experiment, we define a new term \bar{e} which represents the visible part of the object. The \bar{e} is derived by setting the element in e to 1 if its value is obviously smaller than others, while the rest element is set to 0. Based on this term, the observation model using the visible part can be obtained. Given an image state x_t and its observation m_t , the similarity between m_t and the templates is measured by the ℓ_2 -norm of the reconstruction error:

$$s(m_t, T) = \|(m_t - T\hat{\omega})\bar{e}\|_2 \quad (5)$$

where \bar{e} is the complementary set of e , which represents the visible part of the object.

In Equation 5, only the visible part of the object instead of the whole one is adopted for matching. That means this new observation model can automatically focus on the visible part and solve the matching ambiguous problem induced by occlusion. It is shown in [14] that the negative exponential of the reconstruction error is proportional to a Gaussian distribution: $N(m_t\bar{e} : \mu, T\hat{\omega}\bar{e} + \xi)$ as $\xi \rightarrow 0$, where ξ is the Gaussian noise during the observation and μ is the mean. The probability of a state x_t generated from the template is determined as follows:

$$p(m_t|x_t) = N(m_t\bar{e} : \mu, T\hat{\omega}\bar{e} + \xi) \quad (6)$$

2.2 Particle Filter Tracking with Occlusion Reasoning

In this paper, the object is localized with a rectangular window and its state x_t is represented using a six dimension affine parameter $(t_x, t_y, \theta, s, \alpha, \beta)$ where (t_x, t_y) denote the 2-D translation parameters and $(\theta, s, \alpha, \beta)$ are the deforming parameters. For the simplicity, we only analyze the occlusion handling between two objects which can be easily extended to more objects. When severe occlusion happens, there is not enough visible part of the object to provide a reliable

matching. Thus we tackle the mild and severe occlusion separately using different strategies. The occlusion degree is determined by the size of the overlapped region against the whole object.

Deal with Mild Occlusion. Assume $X_t = \{x_t^A, x_t^B\}$ is the state of objects A and B . Given the observation m_t , the goal of tracking is to infer X_t under occlusion. This inference can be cast as a Bayesian posterior inference process:

$$p(X_t, \pi_t | m_t) \propto p(m_t | X_t, \pi_t) \int p(X_t | X_{t-1}) p(X_{t-1} | \pi_{t-1}, m_{t-1}) p(\pi_t | \pi_{t-1}) dX_{t-1} \quad (7)$$

where π_t is the occlusion relationship between A and B .

In the conventional tracking algorithms with occlusion handling, it is imperative to deduce the occlusion relationship between objects. This is because that without the information of π_t , the term $p(m_t | X_t, \pi_t)$ is impossible to derive. However the derivation of the π_t must be based on the object states of previous frame. The inter-determination between the state inference and the occlusion relationship will degenerate the tracking performance and increase the complexity of the algorithm. Also if the deduced occlusion relationship is wrong, the failure in tracking is inevitable. By using the observation model in Equation(6), the occluded part of an object is obtained using the sparse reconstruction under the ℓ_1 -norm constraint. The visible part is automatically explored using the term \bar{e} . The similarity between the image m_t and the templates is measured by using the residual error $\|(m_t \bar{e} - T \hat{\omega} \bar{e})\|_2$. As a result, the occlusion relationship takes no effect on matching the image warped by the object state X_t with the templates. Thus, the Bayesian posterior inference process can be simplified into the following expression:

$$p(X_t | m_t) \propto p(m_t | X_t) \int p(X_t | X_{t-1}) p(X_{t-1} | m_{t-1}) dX_{t-1} \quad (8)$$

With this simplified inference expression, the posterior probability $p(X_t | m_t)$ can be approximated with a set of weighted particles [3]. Given a set of samples $\{x_t^{A,i}, x_t^{B,i}\}_{i=1}^n$ generated from the transition model $p(x_t^A | x_{t-1}^A)$ and $p(x_t^B | x_{t-1}^B)$, the weight of each particle w_t^i is evaluated by the observation likelihood $p(m_t | X_t)$ which is defined as follows:

$$p(m_t | X_t) = p(m_t^A | x_t^{A,k}) p(m_t^B | x_t^{B,l}) \quad (9)$$

where k and l are respectively the k th and l th particle of A, B . The state X_t is obtained by maximizing the weights of the particles.

However if we calculate the weight of each particle in the joints state space, the time complexity is $o(n^p)$, where n and p are respectively the number of particles and objects. If the object number is more than two, the step to calculate the weights of the whole particles will be time consuming. In this paper, a cross iteration procedure instead of directly maximizing the weights of the particles is adopted to increase the efficiency. Assuming $\hat{x}_{t,s}^A, \hat{x}_{t,s}^B$ are the optimal states for the objects A and B at the s th iteration. The tracking problem of object A and B can thus be formulated as follows:

$$\hat{x}_{t,s+1}^A = \arg \max_{x_t^A} p(m_t^A | x_{t,s}^A) p(m_t^B | x_{t,s}^B) \quad (10)$$

$$\hat{x}_{t,s+1}^B = \arg \max_{x_t^B} p(m_t^A | \hat{x}_{t,s+1}^A) p(m_t^B | x_{t,s}^B) \quad (11)$$

The iteration of Equation (10) and (11) continues until convergence. The time complexity of the algorithm decreases from $o(n^p)$ to $o(np)$.

Deal with Severe Occlusion. When more than 70% percent of the object is occluded, there is not enough visible part of the object providing a reliable matching. Also if complete occlusion happens, it becomes impossible to evaluate the observation of the occluded object. To deal with these situations, a new observation model which takes the velocity constraint into consideration is introduced.

Let $\vec{v}_{t-1}^i = x_{t-1}^i - x_{t-2}^i$ and $\vec{v}_t^i = x_t^i - x_{t-1}^i$ be the motion vectors between two consecutive frames. During the tracking process, the motion between two consecutive frames is usually very small. The changes of the object state at time t and $t-1$ will be small accordingly. It is equal to say that the particle moves in constant velocity is favored and set to a larger observation likelihood. Based on the above analysis, the likelihood function is defined as follows:

$$p(m_t | x_t) \propto \exp\{-\Theta_{t,t-1}^v\} \exp\{-\|\vec{v}_t^i - \vec{v}_{t-1}^i\|\} \quad (12)$$

where $\Theta_{t,t-1}^v$ is the angle between \vec{v}_{t-1}^i and \vec{v}_t^i ; $\|\bullet\|$ is the Euclidean norms.

2.3 Update the Template

In order to capture the object appearance changes, an effective mechanism to update the templates is needed. When occlusion happens, we consider the object as a occluded one if the nonzero value in $\bar{\epsilon}$ exceed a certain threshold. The template of the occluded object is not updated to avoid the wrong updating.

Also to overcome the template drifting problem, we introduce an adaptive updating mechanism. The updating process is conducted through the following steps. (1) The template selected by hand in the first frame is kept during the whole tracking process as the stable component. (2) The new tracking result is employed as a new candidate template to be added in the template pool. If the difference between the new template and the template pool is above a threshold, the new obtained template is added into the template pool to reflect the appearance changes. (3) Each template is assigned with a weight which varies over time t . The weight of each template is set as θ^{T-t} , where $\theta \in [0, 1]$, $t \in [1, T]$ is the time of the template being added into the template pool; T is the current time. This strategy is designed to capture the most recently changes. In the experiment, we set the value of θ to be 0.95.

3 Experiments

In order to show the effectiveness of our algorithm in handling occlusion in tracking multi-object, we test it with numerous videos. Comparisons with the state-of-art algorithms are also presented to show the superiority of our approach.

The number of the templates in our experiment is 10. The templates of each object are given at the beginning of the tracking. The first template is initialized by hand and the rest templates are obtained by one pixel displacement of the first one in different direction. During the tracking process, the templates are updated adaptively according to the tracking results.

3.1 Track Multi-object

In order to validate the effectiveness of our algorithm, we test our algorithm on four different videos: two are to track faces and the other two are to track pedestrians. Also the comparisons with several other state-of-art algorithms are presented.

In the first example, two faces occlude each other and the face of one person gradually disappears during the occlusion process. We compare our sparse reconstruction with the incremental subspace algorithm [15]. In [15], an object is represented by a low dimensional eigenspace. The matching process is implemented based on the similarity between the warped image and the subspace spanned by the eigenspace. From the results in Fig.2(b), the influential subspace based matching is not suitable for occlusion handling. Because the occluded part is also used for matching, when the part of the one person disappears, the invisible part can not be reconstructed by the subspace based matching which leads to the failure in tracking. On the contrary, as shown in Fig.2(a), the sparse reconstruction can tackle occlusion and disappearance well. This is because our observation model adopts the visible part for matching, while ignores the occluded part, which is the key of the success in tracking.

In the second example, two faces occlude each other and endure appearance changes. In order to further illustrate the strength of our algorithm, we conduct comparison with two state-of-the-art multi-object tracking algorithms [11,16] which are most similar to our work. In [11], the game-theoretic analysis is introduced to implicitly handle occlusion, while in [16] species competition is used to decompose multi-object tracking with occlusion into single object tracking.

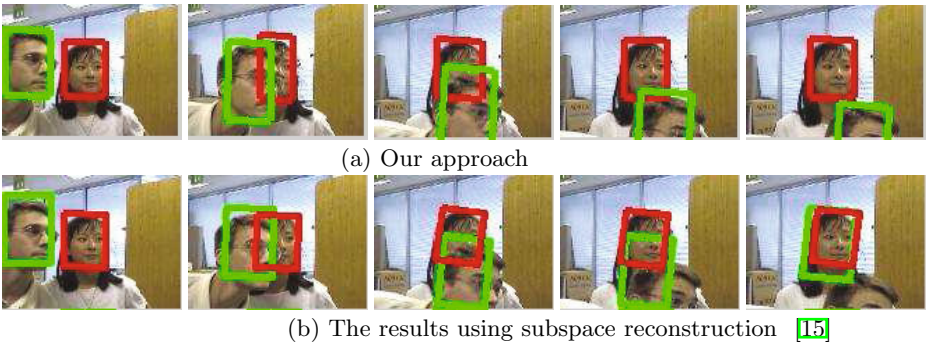


Fig. 2. The results of tracking two occluded faces

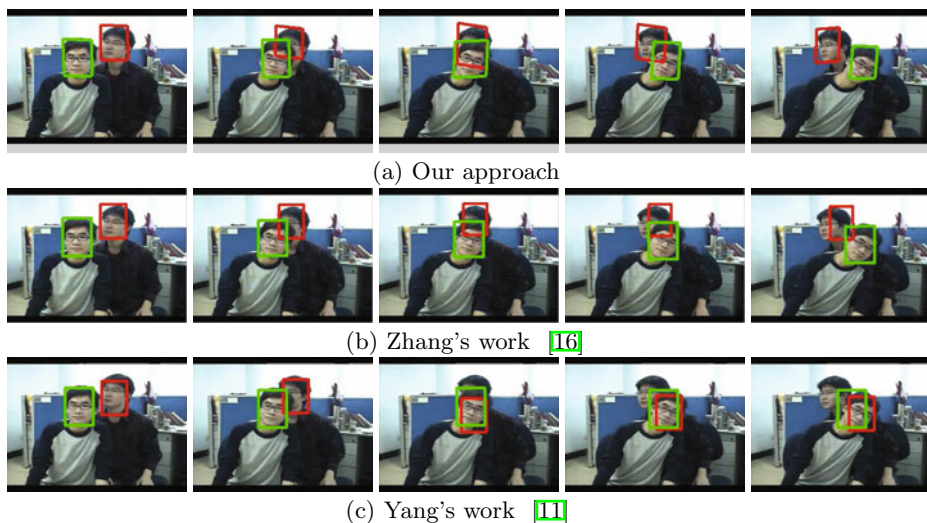


Fig. 3. Two faces occlude each other and endure appearance changes

Both of these two algorithms exclude the step to deduce occlusion relationship via different strategies. The results in Fig.3(b) show that Zhang's algorithm [16] fails to track the occluded object (in the window with red color). The main reason is that the visible part is the most reliable information for the occluded object, while the procedure of species competition can not always output satisfying competition results. From the results in Fig.3(c), Yang's algorithm [11] also can not provide satisfying results. However, the good tracking results in Fig.3(a) illustrate the effectiveness of our algorithm. Also the appearance changes of the faces are effectively handled using our template updating mechanism.

In the third example, we test our algorithm on a video from *PETS2004* which is an open database for visual surveillance, available on <http://homepages.inf.ed.ac.uk/rbf/CAVIAR>. In this video, two men turn around from the side to the front and their appearances change. The woman turns around from the front to the side. Fig.4 illustrates some key frames of the tracking results. As illustrated in Fig.4(a), our method successfully tracks all the pedestrians and effectively handles occlusion. At the same time, the appearance changes of the pedestrians are successfully tackled through our template update strategies. However both in [11,16] the man tracked in green window is distracted by the women tracked with the red. From the results in Fig.4(c), Yang's algorithm [11] quickly loses the object with blue window.

In the last experiment, we test our algorithm on another video from the *PETS* data set in 2006, which is available on <http://pets2006.net/>. Fig.5 illustrates some key frames of the tracking results (Person A is tracked using a blue window, Person B is tracked using a green window, Person C is tracked using a red window). The results in Fig.5(a) show that our algorithm can successfully handle occlusion between different persons, while [11,16] can not deal with occlusion effectively between the persons *B* and *C* which leads to the failure in tracking.

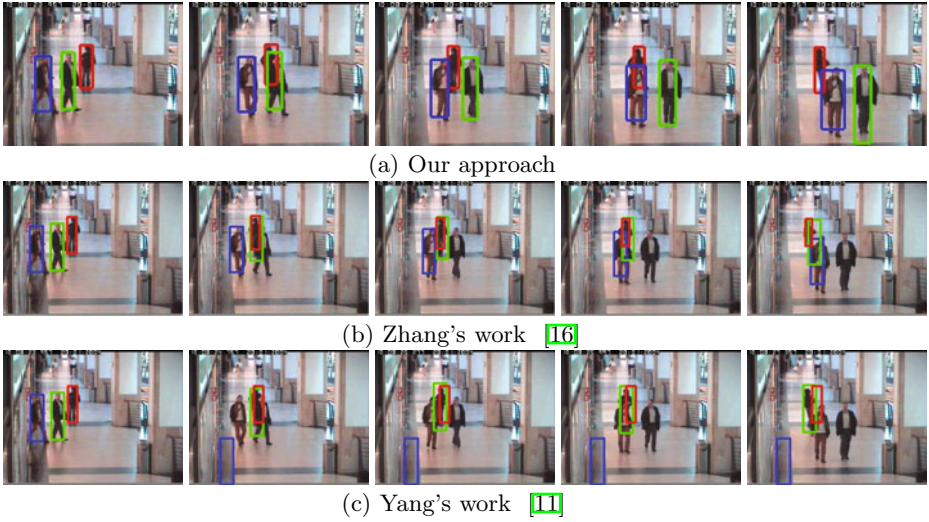


Fig. 4. The tracking results of third sequence

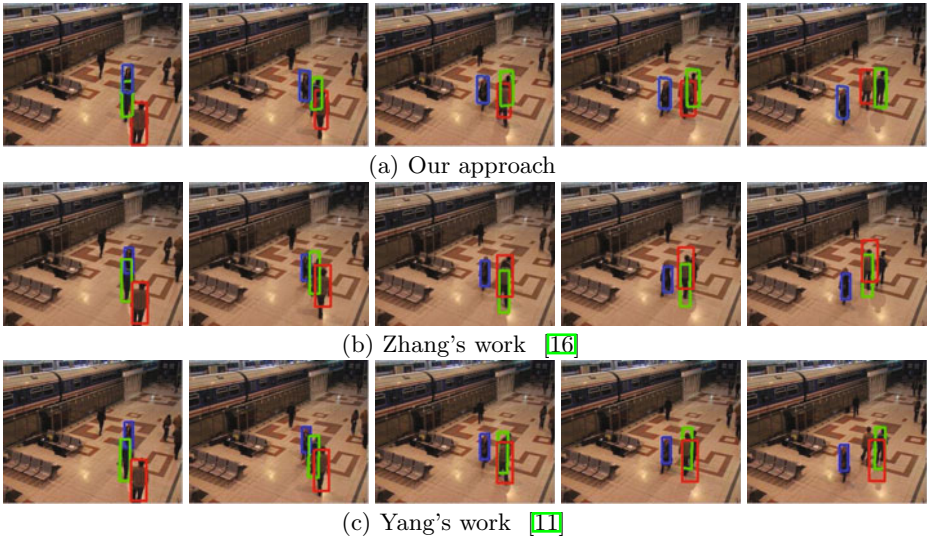


Fig. 5. The tracking results of last sequence

This is because that the certain rules adopted in these two algorithms cannot always obtain the accurate occlusion information, which makes the accurate tracking results cannot always be guaranteed. However in our algorithm, the visible part is obtained from ℓ_1 -regularized sparse reconstruction. As a result, the accurate matching is always available.

Table 1. Quantitative results of our approach, Zhang’s and Yang’s work

Approaches		Our algorithm	Zhang’s work	Yang’s work
Successfully tracked frames	Person A	89/89	89/89	89/89
	Person B	89/89	76/89	57/89
	Person C	89/89	58/89	89/89
RMSE of Position	Person A	3.1545	3.2985	5.6851
	Person B	4.4128	6.9392	17.8217
	Person C	4.2104	14.4789	8.8054

A quantitative evaluation is also given in Table 1 to further demonstrate the superiority of our algorithm. The evaluation is comprised of the following two aspects: the number of successfully tracked frames (the tracking is defined as failure if the center of the window is not in the object), RMSE (root mean square error) between the estimated position and the groundtruth which is obtained by hand. The failure of [11, 16] for the persons *B* and *C* mainly concentrates on the frames when occlusion happens. Additionally, the localization accuracy of our algorithm is apparent superior to those by the other two algorithms.

4 Conclusions

In this paper, we propose an effective framework to track multi-object under occlusion through sparse reconstruction. The matching between the states and the templates is based on the visible part of the occluded objects. In our approach, the deducing of occlusion relationship between the objects or certain rules are not necessary. The tracking of multi-object with occlusion handling becomes a simple joint probability inference problem. Various experiments validate that our approach can successfully handle with occlusion in multi-object tracking.

Acknowledgement. This work is partly supported by NSFC (Grant No. 60825204, 0935002, 60705003 and 61005030) and the National 863 High-Tech R&D Program of China (Grant No. 2009AA01Z318).

References

1. Hager, G., Belhumeur, P.: Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans. on PAMI* 20, 1025–1039 (1998)
2. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. *IEEE Trans. on PAMI* 25, 234–240 (2003)
3. Isard, M., Blake, A.: Condensation: conditional density propagation for visual tracking. *International Journal of Computer Vision* 29, 5–28 (1998)
4. Jepson, A., Fleet, D., El-Maraghi, T.: Robust online appearance models for visual tracking. *IEEE Trans. on PAMI* 25, 1296–1311 (2003)
5. Rasmussen, C., Hager, G.: Probabilistic data association methods for tracking complex visual objects. *IEEE Trans. on PAMI* 23, 560–576 (2001)

6. Elgammal, A., Davis, L.: Probabilistic framework for segmenting people under occlusion. In: Proc. of International Conference on Computer Vision, vol. 2, pp. 145–152 (2001)
7. Wu, Y., Yu, T., Hua, G.: Tracking appearances with occlusions. In: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 89–795 (2003)
8. Sudderth, E., Mandel, M., Freeman, W., Willsky, A.: Distributed occlusion reasoning for tracking with nonparametric belief propagation. In: Advances in Neural Information Processing Systems, pp. 1369–1376 (2004)
9. MacCormick, J., Blake, A.: A probabilistic exclusion principle for tracking multiple objects. *International Journal of Computer Vision* 39, 57–71 (2000)
10. Nguyen, H., Ji, Q., Smeulders, A.: Spatio-temporal context for robust multitarget tracking. *IEEE Trans. on PAMI* 29, 52–64 (2007)
11. Yang, M., Yu, T., Wu, Y.: Game-theoretic multiple target tracking. In: Proc. of ICCV (2007)
12. Wright, J., Yang, A., Ganesh, A., Sastry, S., Ma, Y.: Robust face recognition via sparse representation. *IEEE Trans. Pattern Analysis and Machine Intelligence* 31, 210–227 (2009)
13. Boyd, S., Vandenberghe, L.: *Convex optimization*. Cambridge Univ. Press, Cambridge (2004)
14. Roweis, S.: Em algorithms for pca and spca. In: Advances in Neural Information Processing Systems, vol. 10, pp. 626–632 (1997)
15. Lim, J., Ross, D., Lin, R., Yang, M.: Incremental learning for visual tracking. In: Advances in Neural Information Processing Systems, pp. 793–800 (2004)
16. Zhang, X., Hu, W., Li, W., Qu, W., Maybank, S.: Multi-object tracking via species based particle swarm optimization. In: IEEE International Workshop on Visual Surveillance (2009)

An Approximation Algorithm for Computing Minimum-Length Polygons in 3D Images

Fajie Li and Xiuxia Pan

College of Computer Science and Technology
Huaqiao University, Xiamen, Fujian, China
{li.fajie,panpanty}@hqu.edu.cn

Abstract. Length measurements in 3D images have raised interest in image geometry for a long time. This paper discusses the Euclidean shortest path (ESP) to be calculated in a loop of face-connected grid cubes in the 3D orthogonal grid, which are defined by *minimum-length polygonal* (MLP) *curves*. We propose a new approximation algorithm for computing such an MLP. It is much simpler and easier to understand and to implement than previously published algorithms by Li and Klette. It also has a straightforward application for finding an approximate minimum-length polygonal arc (MLA), a generalization of the MLP problem. We also propose two heuristic algorithms for computing a simple cube-arc within a 3D image component, with a minimum number of cubes between two cubes in this component. This may be interpreted as being an approximate solution to the general ESP problem in 3D (which is known as being NP-hard) assuming a regular subdivision of the 3D space into cubes of uniform size.

1 Introduction

A simple cube-curve g is a loop of face-connected grid cubes in the 3D orthogonal grid; the union \mathbf{g} of those cubes defines the *tube* of g . A *critical edge* of a cube-curve g is such a grid edge which is incident with exactly three different cubes contained in g . This paper discusses Euclidean shortest paths (ESPs) in such tubes, which are defined by *minimum-length polygonal* (MLP) *curves* (see Figure 1, where the red polygon is the MLP while the blue segments are critical edges.).

The general *ESP problem* is as follows: Given it is a Euclidean space which contains (closed) polyhedral obstacles; compute a path which (i) connects two given points in the space, (ii) does not intersect the interior of any obstacle, and (iii) is of minimum Euclidean length. This problem (starting with dimension 3) is known to be NP-hard [5].

3D MLP calculations generalize 2D MLP computations. For example, see [7, 27] for 2D robotics scenarios and [12, 23] for theoretical results. In image analysis shortest curve calculations not only use the Euclidean metric but also use graph metrics; for example, see [26].

3D MLPs calculations are related to the problem of multigrid-convergent length estimation for digitized curves. The length of a simple cube-curve in

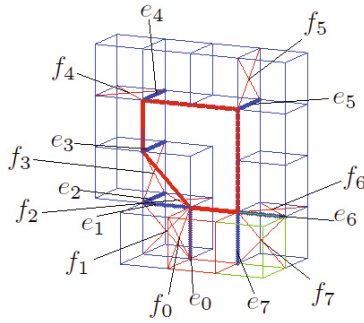


Fig. 1. A simple cube-curve and its MLP

3D Euclidean space can be defined by that of the MLP (see [14, 24, 25]) which is there characterized to be a *global approach* towards length measurement. A *local approach* for 3D length estimation, allowing only weighted steps within a restricted neighborhood, was considered in [10] and [11]. Alternatively to the MLP, the length of 3D digital curves can also be measured (within time linear in the number of grid points on the curve) based on DSS-approximations [6] (DSS = digital straight segment).

3D MLP calculations were first studied by Bülow and Klette [2, 3, 4, 13], they proposed a iterative algorithm called *rubberband algorithm (RBA)* which was experimentally tested and showed “linear run-time behavior” with respect to a pre-selected accuracy constant $\varepsilon > 0$. It proved to be correct for tested inputs, where correctness was possible to be tested manually. However, in those publications, no mathematical proof was given for either linear run time or general convergence (in the sense of approximate algorithms as defined in computational geometry) to the exact solution.

This *original RBA* is also published in the the book [14]. Applications of it are in 3D medical imaging; see, for example, [8, 28]. The correctness and linearity problem of the original RBA was approached by Li and Klette along the following steps:

[15] focused on a very special class of simple cube-curves and proposed a provable correct MLP algorithm which decomposes a cube-curve of that class into arcs at “end angles” (see Definition 3 in [15]). That means the algorithm only applied to the cube-curves which have end-angles.

[16] constructed an example of a simple cube-curve and proved that the MLP of this simple cube-curve does not have any of its vertices at a corner of a grid cube. It follows that any cube-curve with this property does not have any end angle, and the MLP algorithm of [15] cannot be used for all possible inputs. This result showed the existence of cube-curves which require further algorithmic studies.

[20] showed that the original RBA requires a modification (in its Option 3) to guarantee that calculated curves are always contained in the tube \mathbf{g} . This corrected RBA achieves (as the original RBA) a minimization of length by moving vertices along critical edges.

[18] (finally) extended the corrected RBA into an *edge-based RBA* and proved that it is correct for any simple cube-curve. [18] also presented a totally new algorithm, the *face-based RBA*, and showed that it is also correct for any simple cube-curve. It was proved that both, the edge-based and the face-based RBA, have time complexity in $\kappa(\varepsilon) \cdot \mathcal{O}(m)$ time, where m is the number of critical edges in the given simple cube-curve, and

$$\kappa(\varepsilon) = n + (L_0 - L_n)/\varepsilon \tag{1}$$

where L_0 is the length of the initial path, L_n is the length of n -th updated path.

This paper presents a new approximation algorithm which is much simpler and easier to understand and to implement than previously published algorithms by Li and Klette. This paper is organized as follows: Section 2 defines some notations for later usage. Section 3 proposes and discusses the algorithms. Section 4 briefly discusses the correctness and time complexity of the algorithms. Section 5 presents the experimental results of the proposed algorithms. Section 6 concludes.

2 Basics

We use definitions and results from [19]: An arithmetic algorithm is *eventually exact* if it provides also final (not necessarily arithmetic) steps for converting its approximate solution into the true solution (Definition 3, page 5).

Theorem 1. (Corollary 4, page 97) *There does not exist an exact algorithm for calculating the MLP of any simple cube-curve.*

Let c_p and c_q be two cubes in the same connected component, $A(c_p, c_q)$ (see on the right of Figure 2) an arc between two cubes c_p and c_q , and $|A(c_p, c_q)|$ the number of cubes contained in the arc $A(c_p, c_q)$.

Let MLPP denote the class of any minimum-length polygonal curve problem. We may generalize the problems in MLPP to *minimum-length polygonal arc* problems as follows: Let $p \in c_p$ and $q \in c_q$, compute the shortest path between p and q inside of $A(c_p, c_q)$. We denote this generalized class of problems by MLAP. For example, given it is a simple cube-arc as shown on the right of Figure 2 and two points p and q . Section 7.4 in [19] proves that there does not exist an exact algorithm for calculating the minimum-length polygonal arc from p to q inside of the arc.

In this paper we propose a new simple face-based rubberband algorithm for computing approximately the MLP. Our algorithm has also a straightforward application for finding approximation solutions to problems in MLAP. We also present two heuristic algorithms for computing simple cube-arcs, each with a minimum number of cubes, between two cubes in the same connected component. Combined with the MLAP algorithm, this provides an approximate solution to the general ESP problem in 3D (which is NP-hard as mentioned above [5]) when subdividing the 3D space into uniformly sized cubes.

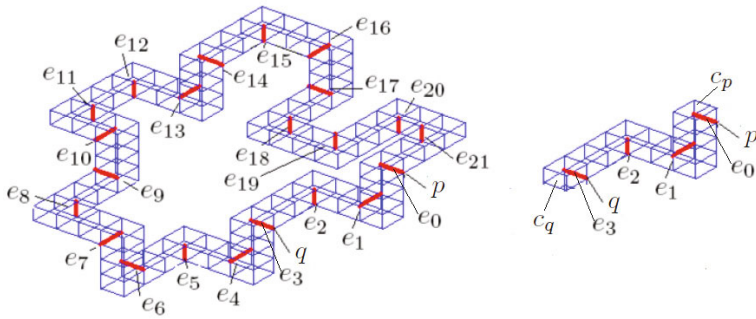


Fig. 2. A simple cube-arc (*right*) as a part of a simple cube-curve (*left*)

We also use some definitions from [18]. If f is a face of a cube in g and one of f 's edges is a critical edge e in g then f is called a *critical face of e in g* , or (for short) a *critical face*. Let e be a critical edge of a simple cube-curve g and f_1, f_2 be two critical faces of e in g . Let c_1, c_2 be the centers of f_1, f_2 respectively. Then a polygonal curve can go in the direction from c_1 to c_2 , or from c_2 to c_1 , to visit all cubes in g such that each cube is visited exactly once. If e is on the left of line segment c_1c_2 , then the orientation from c_1 to c_2 is called *counter-clockwise orientation* of g . f_1 is called *the first critical face of e in g* . If e is on the right of line segment c_1c_2 , then the direction from c_1 to c_2 is called *clockwise orientation* of g . $d_e(p, q)$ denotes the Euclidean distance between two points p and q .

Figure 1 shows all critical edges ($e_0, e_1, e_2, \dots, e_7$) and their first critical faces ($f_0, f_1, f_2, \dots, f_7$) of a simple cube-curve, denoted by g_8 . Let s_i and s'_i be i -th side of faces f and f' respectively ($i = 1, 2, 3, 4$). If f contains f' and the Euclidean distance between s_i and s'_i is ε ($i = 1, 2, 3, 4$), then we say that f' is obtained from f by ε -dilation, or, in short, a (first critical) dilation face (see Figure 3).

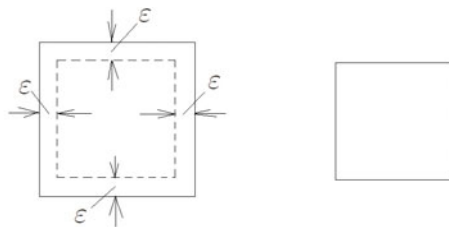


Fig. 3. Illustration of ε -dilation. *Left*: a first critical face. *Right*: a first critical ε -dilation face.

Recall the following definition; see, for example, [9]: An algorithm is an δ -approximation algorithm for a minimization problem P iff, for each input of P , the algorithm delivers a solution that is at most δ times the optimum solution. Corresponding to the definition of δ -approximation algorithms, we introduce the following definition: A MLP is a δ -approximation (Euclidean) closed path for an MLP problem iff its length is at most δ times the optimum solution. Let $f_0, f_1,$

..., and f_{k-1} be k (all) continuous first critical faces or first critical dilation face ($k \geq 2$) in g , $p \in f_0$, and $q \in f_{k-1}$. Let $L_g(p, q)$ be the length of the shortest path, starting at $p \in f_0$, then visiting faces or dilation faces f_1, \dots, f_{k-2} and $q \in f_{k-1}$ in order, and finally ending at $p \in f_0$. Let S_0, S_1, \dots, S_{k-1} be k non-empty sets; let $\prod_{i=0}^{k-1} S_i$ be the cross product of those sets.

Suppose that the side of each cube has length of 1, then each cube can be defined by a corner of it. Let f_c be the front face of a cube c and $v_c = (x_c, y_c, z_c)$ the left bottom vertex of f_c . Then c can be defined by v_c . v_c is called the *defining vertex* of c (see Figure 4).

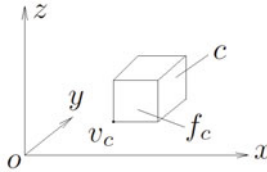


Fig. 4. The defining vertex of a cube

Let c_s and c_t be two cubes in the same component \mathcal{C} . $c_s = (x_s, y_s, z_s)$, $c_t = (x_t, y_t, z_t)$. Denote

$$d(c_s, c_t) = |x_t - x_s| + |y_t - y_s| + |z_t - z_s| + 1 \tag{2}$$

Let $N(c_s)$ be the set of 6-neighbors of c_s which consists of at most six cubes such that each of them is in \mathcal{C} . We partition $N(c_s)$ into three subsets $N_i(c_s)$ ($i = 1, 2,$ and 3), where $N_1(c_s) = \{c : (c = c_s + (x_t - x_s)/|x_t - x_s|, 0, 0) \vee (c = c_s + (0, y_t - y_s)/|y_t - y_s|, 0) \vee (c = c_s + (0, 0, z_t - z_s)/|z_t - z_s|) \wedge (c \in N(c_s))\}$; $N_2(c_s) = \{c = (x_c, y_c, z_c) : (x_c = x_t) \vee (y_c = y_t) \vee (z_c = z_t) \wedge (c \in N(c_s))\}$; and $N_3(c_s) = \{c : (c = c_s - (x_t - x_s)/|x_t - x_s|, 0, 0) \vee (c = c_s - (0, y_t - y_s)/|y_t - y_s|, 0) \vee (c = c_s - (0, 0, z_t - z_s)/|z_t - z_s|) \wedge (c \in N(c_s))\}$. In other words, $N_1(c_s)$ consists of cubes which are located closer to c_t than c_s ; $N_2(c_s)$ consists of cubes which have at least one coordinate equal to that of c_t ; $N_3(c_s)$ consists of cubes which are located further to c_t than c_s .

3 Algorithms

In this section we start presenting the main algorithm for efficiently computing approximate MLP. Then we describe two heuristic algorithms for finding a shortest cubic arc between two cubes in a connected component.

3.1 The Algorithm for Computing an Approximate MLP

The first difficult task for applying a rubberband algorithm (RBA), for example, as shown in Option 2 in [4], is to find the so-called “step set”. Another issue when applying a RBA is to deal with the degenerative case of the RBA. The following algorithm overcomes the first difficulty by simply taking all the initial critical faces as the step set. It handles the second task by ε_2 -dilation.

- 1 Approximation MLP Algorithm
- 2 *Input:* k first critical faces f_0, f_1, \dots, f_{k-1} , and two chosen accuracy constants ε_1 and ε_2 .
- 3 *Output:* An updated closed $\{1 + 4k \times [r(\varepsilon_1) + \sqrt{2} \times \varepsilon_2]/L\}$ -approximation path (MLP) $\rho(s, p_0, \dots, p_1, \dots, p_{k-1}, s)$, which may also contain vertices of Π , where L is the length of an optimal path, $r(\varepsilon_1)$ the upper error bound [1] for distances between p_i and the corresponding optimal vertex p'_i : $d_e(p_i, p'_i) \leq r(\varepsilon_1)$, for $i = 0, 1, \dots, k - 1$.

- 1: For each $i \in \{0, 1, \dots, k - 1\}$, update face f_i by ε_2 -dilation; let p_i be the center of f_i ; let L_0 be $\sum_{i=0}^{k-1} d_e(p_i, p_{i+1})$ (all subscripts take mod k); and L_1 be ∞ .
- 2: **while** $L_1 - L_0 > \varepsilon_1$ **do**
- 3: **for** each $i \in \{0, 1, \dots, k - 1\}$ **do**
- 4: Compute $q_k \in f_k$ such that (see Figure 5) $d_e(p_{k-1}, q_k) + d_e(q_k, p_{k+1}) = \min\{d_e(p_{k-1}, q) + d_e(q, p_{k+1}) : q \in f_k\}$; update ρ by replacing p_k by q_k .
- 5: **end for**
- 6: Let L_0 be L_1 ; calculate the perimeter L_1 of ρ .
- 7: **end while**
- 8: Output ρ and the desired length equals to L_1 .

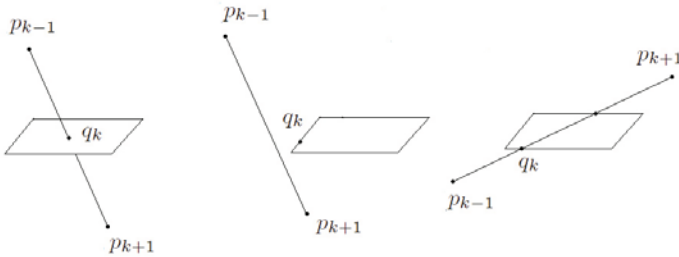


Fig. 5. Illustration for Step 4 in this MLP algorithm

3.2 Algorithms for Finding Arcs with Minimal Number of Cubes

The first algorithm is inspired by the flow of rivers which always go along the shortest path among the obstacles. The second one is a refined version of the first one. The third one is inspired by isometric search.

- 1 Flow Algorithm
- 2 *Input:* Let c_p and c_q be two cubes in the same component \mathcal{C} .
- 3 *Output:* An arc $A(c_p, c_q)$ inside of the same connected component \mathcal{C} .

```

1: Put  $c_p$  into a stack.
2: while The stack is not empty do
3:   Pop  $c$  out of the stack.
4:   if  $c = c_q$  then
5:     Stop.
6:   else
7:     Partition  $N(c)$  into three subsets:
        $N(c) = N_1(c) \cup N_2(c) \cup N_3(c)$ .
8:     Put the cubes in  $N_3(c)$ ,  $N_2(c)$  and  $N_1(c)$  into the stack.
9:   end if
10: end while

```

1 Refined Flow Algorithm

2 Input: Let c_p and c_q be two cubes in the same component \mathcal{C} .

3 Output: A shortest arc $A(c_p, c_q)$.

Each cube is combined with a non-negative integer $size(c_p)$ which is the number of cubes from the starting cube c_p to the current one, and also combined with its parent cube $c_m(c_p)$ (m is short for mother.)

The main idea of this algorithm is straightforward: Apply Flow Algorithm (Algorithm 3) to obtain an initial arc $A(c_p, c_q)$. If $|A(c_p, c_q)| = d(c_p, c_q)$, then output $A(c_p, c_q)$ and stop. Otherwise, update $A(c_p, c_q)$ such that $|A(c_p, c_q)|$ is decreased. Repeat this procedure until $|A(c_p, c_q)|$ can not be decreased.

```

1: To initialize, let the  $size(c_p)$  be 0, the  $size(c)$  of each cube  $c$  in  $N_i(c_p)$  be
   1 ( $i = 1, 2,$  and  $3$ ),  $c_m(c) = c_p$ . For each cube  $c' \in \mathcal{C} \setminus (N_i(c_p) \cup \{c_p\})$ , let
    $size(c')$  be -1.
2: Put the cubes in  $N_3(c_p)$ ,  $N_2(c_p)$  and  $N_1(c_p)$  into a stack.
3: while The stack is not empty do
4:   Pop  $c$  out of the stack.
5:   if  $c = c_q$  then
6:     if  $|A(c_p, c_q)| = d(c_p, c_q)$  then
7:       Stop.
8:     else
9:       Let  $len(c) = |A(c_p, c_q)|$ .
10:    end if
11:   end if
12:   for each  $c' \in N(c)$  do
13:     if  $size(c') = -1$  then
14:        $size(c') = size(c) + 1$ .
15:     else
16:       if  $len(c') > size(c')$  then
17:          $size(c') = size(c) + 1$ ;  $c_m(c') = c$ .
18:       end if
19:     end if

```

```

20:   Partition  $N(c)$  into three subsets:
        $N(c) = N_1(c) \cup N_2(c) \cup N_3(c)$ .
21:   Put the cubes in  $N_3(c)$ ,  $N_2(c)$  and  $N_1(c)$  into the stack.
22:   end for
23: end while

```

```

1 Isometric Extension Algorithm
2 Input: Let  $c_p$  and  $c_q$  be two cubes in the same component  $\mathcal{C}$ .
3 Output: A shortest arc  $A(c_p, c_q)$ .

```

```

1: To initialize, let the  $size(c_p)$  be 0, the  $size(c)$  of each cube  $c$  in  $N(c_p)$  be 1,
    $c_m(c) = c_p$ . For each cube  $c' \in \mathcal{C} \setminus (N(c_p) \cup \{c_p\})$ , let  $size(c')$  be -1.
2: while true do
3:   Pop  $c$  out of the queue.
4:   if  $c = c_q$  then
5:     Return  $A(c_p, c_q)$  and its length  $size(c) + 1$ .
6:   else
7:     for each  $c' \in N(c)$  do
8:       if  $size(c') = -1$  then
9:          $c_m(c') = c$ ;  $size(c') = size(c) + 1$ ; Put  $c'$  in the queue.
10:      end if
11:    end for
12:  end if
13: end while

```

A correctness proof of Algorithms 3–3 is straightforward. Their time complexity equals $\mathcal{O}(n)$, where n is the number of cubes in the connected component \mathcal{C} .

4 Correctness and Time Complexity

We apply basic results of convex analysis; see, for example, [1, 21, 22]:

Theorem 2. ([22], Theorem 3.5) *Let S_1 and S_2 be convex sets in \mathbb{R}^m and \mathbb{R}^n , respectively. Then $S_1 \times S_2$ is a convex set in \mathbb{R}^{m+n} , where $m, n \in \mathbb{N}$.*

Proposition 1. *Each norm on \mathbb{R}^n is a convex function ([1], page 72); a non-negative weighted sum of convex functions is a convex function ([1], page 72).*

Proposition 2. ([22], page 264) *Let f be a convex function. If x is a point where f has a finite local minimum, then x is a point where f has its global minimum.*

Our results are as follows:

Proposition 3. *Each face or dilated face is a convex set.*

By Theorem 2 and Propositions 1 and 3, we have the following

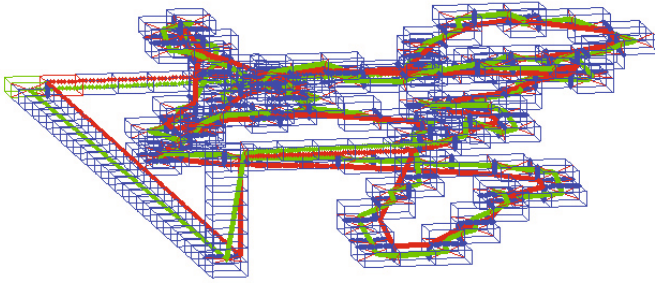


Fig. 6. Illustration of the results of Algorithm 3

Table 1. Resulting data obtained from Algorithms 3: i and i' are the indices of experiment; m and m' the numbers of critical edges; I and I' the numbers of iterations taken; L_0 and L'_0 the lengths of initial paths; L and L' the lengths of resulting paths; $\delta = L_0 - L$; and $\delta' = L'_0 - L'$

i	m	I	L_0	L	δ	i'	m'	I'	L'_0	L'	δ'
1	13	37	19·35	15·85	3·49	1	12	1559	19·73	15·59	4·14
2	19	30	29·40	24·72	4·69	2	19	1505	33·35	26·99	6·36
3	26	27	45·02	38·97	6·04	3	25	3832	42·94	35·04	7·90
4	33	25	54·49	46·58	7·91	4	36	1674	43·99	35·57	8·42
5	40	34	46·25	36·53	9·72	5	40	3610	58·00	46·84	11·16
6	48	38	69·34	57·02	12·32	6	48	5877	75·52	64·13	11·39
7	54	92	79·30	67·67	11·63	7	59	1831	78·29	62·95	15·34
8	58	22	103·61	87·29	16·32	8	64	2127	106·23	88·28	17·95
9	74	48	103·57	88·49	15·08	9	69	1777	88·33	68·27	20·06
10	78	81	95·75	78·38	17·37	10	81	2281	116·83	94·37	22·46

Corollary 1. $L_g(p, q): f_0 \times f_1 \times \dots \times f_{k-1} \times f_0 \rightarrow \mathbb{R}$ is a convex function.

Theorem 3. If the chosen accuracy constant ε is sufficiently small, then Algorithm 3 outputs an $\{1 + 4k \times [r(\varepsilon_1) + \sqrt{2} \times \varepsilon_2]/L\}$ -approximation global MLP.

Proof. By Propositions 2, Algorithm 3 outputs an approximation global MLP. For each $i \in \{1, 2, \dots, k-1\}$, the error of the difference between $d_e(p_i, p_{i+1})$ and $d_e(v_i, v_{i+1})$ is at most $4 \times r(\varepsilon_1) + \sqrt{2} \times \varepsilon_2$ because of $d_e(p_i, v_i) \leq r(\varepsilon) + \sqrt{2} \times \varepsilon_2$. We obtain that

$$\begin{aligned}
 L &\leq \sum_{i=0}^{k-1} d_e(p_i, p_{i+1}) \leq \sum_{i=0}^{k-1} [d_e(v_i, v_{i+1}) + 4 \times r(\varepsilon_1) + \sqrt{2} \times \varepsilon_2] \\
 &= L + 4k \times [r(\varepsilon_1) + \sqrt{2} \times \varepsilon_2]
 \end{aligned}$$

Thus, the output path is an $\{1 + 4k \times [r(\varepsilon_1) + \sqrt{2} \times \varepsilon_2]/L\}$ -approximation path. This proves the theorem. \square

Regarding the time complexity of our solution to the approximation MLP, we state that the main computation is in the two stacked loops. The while-loop takes $\kappa(\varepsilon_1)$ iterations; the for-loop can be computed in time $\mathcal{O}(k)$. Thus, Algorithm 3 can be computed in time

$$\kappa(\varepsilon_1) \cdot \mathcal{O}(k) \tag{3}$$

We may conclude that this paper provided an $\{1 + 4k \times [r(\varepsilon_1) + \sqrt{2} \times \varepsilon_2]/L\}$ -approximation solution to the approximation MLP, having time complexity $\kappa(\varepsilon_1) \cdot \mathcal{O}(k)$, where k is the number of the first critical faces, and L is the length of an optimal MLP.

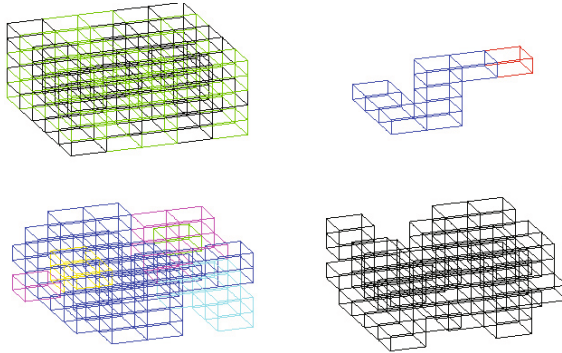


Fig. 7. Illustration of the results of Algorithm 3: Top left shows both cubes in a volume image and in the background; bottom left cubes in the volume image ($8 \times 8 \times 8$); bottom right cubes in the background; top right the shortest arc in a connected component

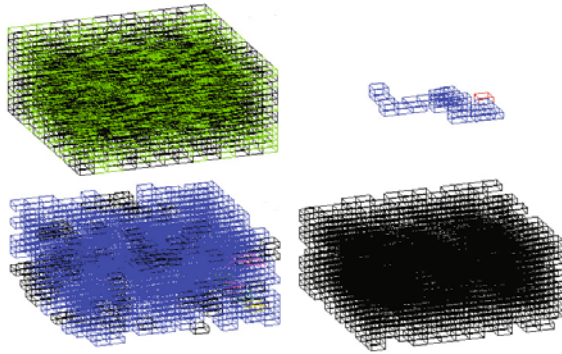


Fig. 8. Illustration of analogous results of Algorithm 3 in the volume image ($15 \times 15 \times 15$)

5 Experimental Results

Figure 6 shows the resulting MLP (in red) obtained by Algorithm 3 when both chosen accuracy constants ε_1 and ε_2 are set to be 10^{-6} and 10^{-3} , respectively. The initial path is in green. Table 1 shows the difference in the numbers of

iterations taken in Algorithm 3 when the first accuracy constant ε_1 was set to be 10^{-6} while the second accuracy constant ε_2 was set to be 10^{-3} or 10^{-1} .

Both Figures 7 and 8 show the results of Algorithm 3.

6 Concluding Remarks

We propose a new simple approximation algorithm for computing MLPs in 3D space. Experimental results show that the iteration number of the algorithm is very sensible to the second chosen accuracy constant ε_2 when the first chosen accuracy constant ε_1 is fixed. The algorithm has applications for finding approximate solutions to MLAPs, which generalize the MLP problem. Arc length problems are today of relevance in 3D medical imaging (e.g., brain cell or lung tissue analysis) or in 3D crystal imaging, just to mention two examples.

We also proposed and implemented two heuristic algorithms for computing simple cube-arcs with minimum numbers of cubes between two cubes. Our algorithm may find an approximate solution to the general ESP in 3D which is known to be NP-hard.

Acknowledgement. The authors thank Reinhard Klette for advice on this research.

References

1. Boyd, S., Vandenberghe, L.: *Convex Optimization*. Cambridge University Press, Cambridge (2004)
2. Bülow, T., Klette, R.: Rubber band algorithm for estimating the length of digitized space-curves. In: *Proc. Intern. Conf. Pattern Recognition*, vol. 3, pp. 551–555 (2000)
3. Bülow, T., Klette, R.: Approximation of 3D shortest polygons in simple cube curves. In: Bertrand, G., Imiya, A., Klette, R. (eds.) *Digital and Image Geometry*. LNCS, vol. 2243, pp. 281–294. Springer, Heidelberg (2002)
4. Bülow, T., Klette, R.: Digital curves in 3D space and a linear-time length estimation algorithm. *IEEE Trans. Pattern Analysis Machine Intelligence* 24, 962–970 (2002)
5. Canny, J., Reif, J.H.: New lower bound techniques for robot motion planning problems. In: *Proc. IEEE Conf. Foundations Computer Science*, pp. 49–60 (1987)
6. Coeurjolly, D., Debled-Rennesson, I., Teytaud, O.: Segmentation and length estimation of 3D discrete curves. In: Bertrand, G., Imiya, A., Klette, R. (eds.) *Digital and Image Geometry*. LNCS, vol. 2243, pp. 299–317. Springer, Heidelberg (2002)
7. Dror, M., Efrat, A., Lubiw, A., Mitchell, J.: Touring a sequence of polygons. In: *Proc. STOC*, pp. 473–482 (2003)
8. Ficarra, E., Benini, L., Macii, E., Zuccheri, G.: Automated DNA fragments recognition and sizing through AFM image processing. *IEEE Trans. Inf. Technol. Biomed.* 9, 508–517 (2005)
9. Hochbaum, D.S.: *Approximation Algorithms for NP-Hard Problems*. PWS Pub. Co., Boston (1997)
10. Jonas, A., Kiryati, N.: Length estimation in 3-D using cube quantization. In: *Proc. Vision Geometry*. SPIE, vol. 2356, pp. 220–230 (1994)
11. Jonas, A., Kiryati, N.: Length estimation in 3-D using cube quantization. *J. Math. Imaging Vision* 8, 215–238 (1998)

12. Karavelas, M.I., Guibas, L.J.: Static and kinetic geometric spanners with applications. In: Proc. ACM-SIAM Symp. Discrete Algorithms, pp. 168–176 (2001)
13. Klette, R., Bülow, T.: Minimum-length polygons in simple cube-curves. In: Nyström, I., Sanniti di Baja, G., Borgfors, G. (eds.) DGCI 2000. LNCS, vol. 1953, p. 467. Springer, Heidelberg (2000)
14. Klette, R., Rosenfeld, A.: Digital Geometry. Morgan Kaufmann, San Francisco (2004)
15. Li, F., Klette, R.: Minimum-length polygon of a simple cube-curve in 3D space. In: Klette, R., Žunić, J. (eds.) IWCIA 2004. LNCS, vol. 3322, pp. 502–511. Springer, Heidelberg (2004)
16. Li, F., Klette, R.: The class of simple cube-curves whose mLPs cannot have vertices at grid points. In: Andrès, É., Damiand, G., Lienhardt, P. (eds.) DGCI 2005. LNCS, vol. 3429, pp. 183–194. Springer, Heidelberg (2005)
17. Li, F., Klette, R.: Minimum-length polygons of first-class simple cube-curves. In: Gagalowicz, A., Philips, W. (eds.) CAIP 2005. LNCS, vol. 3691, pp. 321–329. Springer, Heidelberg (2005)
18. Li, F., Klette, R.: Shortest paths in a cuboidal world. In: Reulke, R., Eckardt, U., Flach, B., Knauer, U., Polthier, K. (eds.) IWCIA 2006. LNCS, vol. 4040, pp. 415–429. Springer, Heidelberg (2006)
19. Li, F., Klette, R.: Exact and approximate algorithms for the calculation of shortest paths. IMA Minneapolis (2006) Report 2141 on, <http://www.ima.umn.edu/preprints/oct2006>
20. Li, F., Klette, R.: Analysis of the rubberband algorithm. Image and Vision Computing 25, 1588–1598 (2007)
21. Roberts, A.W., Varberg, V.D.: Convex Functions. Academic Press, New York (1973)
22. Rockafellar, R.T.: Convex Analysis. Princeton University Press, Princeton (1970)
23. Sklansky, J., Kibler, D.F.: A theory of nonuniformly digitized binary pictures. IEEE Trans. Systems Man Cybernetics 6, 637–647 (1976)
24. Sloboda, F., Zařko, B., Klette, R.: On the topology of grid continua. In: Proc. Vision Geometry, SPIE, vol. 3454, pp. 52–63 (1998)
25. Sloboda, F., Zařko, B., Stoer, J.: On approximation of planar one-dimensional grid continua. In: Klette, R., Rosenfeld, A., Sloboda, F. (eds.) Advances in Digital and Computational Geometry, pp. 113–160. Springer, Heidelberg (1998)
26. Sun, C., Pallottino, S.: Circular shortest path on regular grids. CSIRO Math. Information Sciences, CMIS Report No. 01/76, Australia (2001)
27. Talbot, M.: A dynamical programming solution for shortest path itineraries in robotics. Electr. J. Undergrad. Math. 9, 21–35 (2004)
28. Wolber, R., Stäb, F., Max, H., Wehmeyer, A., Hadshiew, I., Wenck, H., Rippke, F., Wittern, K.: Alpha-Glucosylrutin: Ein hochwirksams Flavonoid zum Schutz vor oxidativem Stress. J. German Society Dermatology 2, 580–587 (2004)

Classifier Acceleration by Imitation

Takahiro Ota^{1,2}, Toshikazu Wada¹, and Takayuki Nakamura¹

¹ Wakayama University No. 930 Sakaedani, Wakayama-shi, Wakayama,
640-8510 Japan

² Kishugikenkogyo Co., Ltd No. 466 Nunobiki, Wakayama-shi, Wakayama,
641-0015 Japan

tohta@kishugiken.co.jp, twada@ieee.org, ntakayuk@sys.wakayama-u.ac.jp

Abstract. This paper presents a framework named “Classifier Molding” that imitates arbitrary classifiers by linear regression trees so as to accelerate classification speed. This framework requires an accurate (but slow) classifier and large amount of training data. As an example of accurate classifier, we used the Compound Similarity Method (CSM) for Industrial Ink Jet Printer (IJJP) character recognition problem. The input-output relationship of trained CSM is imitated by a linear regression tree by providing a large amount of training data. For generating the training data, we developed a character pattern fluctuation method simulating the IJJP printing process. The learnt linear regression tree can be used as an accelerated classifier. Based on this classifier, we also developed Classification based Character Segmentation (CCS) method, which extracts character patterns from an image so as to maximize the total classification scores. Through extensive experiments, we confirmed that imitated classifiers are 1500 times faster than the original classifier without dropping the recognition rate and CCS method greatly corrects the segmentation errors of bottom-up segmentation method.

1 Introduction

Industrial Ink Jet Printers (IJJPs) are widely used in production lines for product marking. IJJPs can work in dusty, dry, and/or high-temperature product lines. These conditions may cause nozzle clogging. Also, IJJP operators may input incorrect information. Both of them cause deteriorated or unwanted printing, which should be immediately detected and fixed. For detecting them, a high-speed and accurate optical character recognition (OCR) system is required.

There are varieties of OCR algorithms; some are fast but inaccurate and the others are accurate but slow. This situation is common, because those classifiers having strong discriminant power waste computational resources and computationally efficient classifiers tend to be inaccurate. For these reasons, it is difficult to construct a classifier which is fast and accurate. Our framework to solving this problem is simple: “*imitate the behavior of an accurate but slow classifier by a simple and fast computational mechanism*”. We call this framework “Classifier Molding”. This consists of the following stages: 1) a classifier is learnt by using labeled training data, 2) a flexible learner imitates the behavior of the learnt

classifier by copying the input-output relationship, 3) the original classifier is replaced by trained learner, and the classification speed improved.

This imitation can be regarded as a nonlinear regression. Among lots of nonlinear regression methods, we employ the Linear Regression Tree [1] in this paper. This is because it has two advantages: it is both fast and flexible.

As a computational model, linear regression tree is faster than standard classifiers, because it performs two simple computations: 1) a binary tree search based on input vector value and 2) the computes product between input vector and matrix stored in the reached leaf node.

In the learning stage, the linear regression tree performs linear regression and domain (input space) decomposition recursively until the regression error becomes smaller than given threshold. Through this recursive decomposition and linear regression, the binary search tree and the regression matrices are obtained. This property guarantees the flexibility that any functions can be well approximated.

This flexibility, however, requires dense training data in the learning stage, because the flexibility implies a poor generalization property as a learning mechanism. This property can be compensated only by providing dense training data spreading in the domain. Usually, the training data size of a linear regression tree should be bigger than that of original classifier. This implies that the training data of a linear regression tree should be generated from that of original classifier.

According to the framework ‘‘Classifier Molding’’ described above, this paper shows an example realization for IIJP-OCR.

1. As an example of an accurate but slow classifier, we selected CSM (Compound Similarity Method) [2] based on our benchmark experiment.
2. As an example of training data multiplication, we present a pattern fluctuation method of training data by simulating the IIJP printing process.

Based on these components, an accurate and fast classifier is realized.

We also developed Classification based Character Segmentation (CCS) method that extracts character image segments from an image so as to maximize the total classification score. The bottom-up segmentation and classification approach cannot deal with the character segmentation errors caused by touching characters; in contrast, CCS corrects the segmentation errors by examining multiple sequences of image segments and finds the sequence having the maximum classification score. Since CCS classifies multiple image segment sequences, it requires a fast classifier. This paper shows that a segmentation-robust character recognition system realized by combining ‘‘Classifier Molding’’ and CCS.

In the following sections, related works in Section 2, the classifier molding framework for IIJP is proposed in Section 3, a CCS method is proposed in Section 4, and Section 5 presents experimental results.

2 Related Works

In many classification tasks, the accuracy and speed of a classifier are incompatible. For example, kernel SVM (Support Vector Machine) [3] can create a non-linear classification boundary, which is supported by many support vectors.

In the classification stage, we have to compute the inner products of an input vector with the support vectors. As the number of inner products increases, the classification accuracy increases, however, the speed slows down.

Our idea breaking the incompatibility is to imitate the behavior of an accurate classifier by a regression tree. The original regression tree is proposed by Breiman [4] et al. as a function approximation technique, which consists of binary search tree and output values stored in the leaf nodes. In this method, the domain is divided into sub-domains so that the output values can be well approximated by constant values. Because of this mechanism, the height of the tree becomes big when performing complex function regression and a smooth output function is approximated by stepwise values. Quinlan [1] extended the regression tree so as to perform the linear regression using regression coefficients stored in the leaf nodes. This extension solves these problems.

In linear regression tree, regression errors become smaller in the earlier division stage. This makes the tree height shorter. Therefore, the time consumption in binary search is also shorter than the original regression tree. The output computation time in the linear regression tree is longer than the original regression tree, because linear regression requires a weighted sum of the input vector with the regression coefficients. This approach is widely accepted by many researchers, and they tried to improve the linear regression tree, because Quinlan's method employs a space splitting criterion by thresholding the variance of the data in a sub-domain corresponding to a node, which is not suitable for some applications.

Karalic [5] proposed a method for finding the best splitting position of an axis so that the sum of the errors of two regressions is minimized. This is done by a brute force manner: errors are computed while sliding the splitting position and find the optimal position. The drawback of this method is long computational time, because this method applies the linear regression expressions for all possible splitting positions. Alexander [6] et al. proposed an efficient algorithm for one dimensional domain. Chaudhari [7] proposed a method to select a regression model for each sub-domain. The regression models are constant value, linear, and higher-order polynomials. Also, he proposed a method to determine the splitting position by using the sign of the regression errors. Dora [8] et al. proposed a method to determine the splitting position by using EM algorithm.

These space splitting methods can be classified into two types, fast splitting but poor accuracy and slow splitting but high accuracy. For solving the problem, Nakamura et al. [9] proposed a PaLM-tree (Partially Linear Mapping tree) employing the Split-and-Merge strategy for domain decomposition similar with the method used in image segmentation [10]. The PaLM-tree also has two advantages: high-dimensional input and output vectors can be handled, and dimensionality reduction mechanism is embedded for avoiding multi-collinearity problem.

3 Classifier Molding by Linear Regression Tree

Most classifiers produce similarity measures or a posteriori probabilities for the combination of an input and a class. Suppose that $\mathbf{x} = (x_1, \dots, x_p)$ represents

an input vector, and $\Omega = (\omega_1, \dots, \omega_q)$ represents classes, then the function of a classifier can be modeled by a mapping f from \mathbf{x} to $\mathbf{y} = (P(\omega_1|x), \dots, P(\omega_q|x))$. By selecting the maximum element in the output vector \mathbf{y} , we can classify \mathbf{x} .

The idea of Classifier Molding is to learn the mapping from \mathbf{x} to \mathbf{y} and utilize the learnt model as a classifier for acceleration. As a mapping learner, we employ linear regression tree in this paper.

Linear Regression Tree. In this section, we describe the construction algorithm using the following notations.

Input dataset: $X = \{\mathbf{x}_1, \dots, \mathbf{x}_p\}$, Classifier output dataset: $Y = \{\mathbf{y}_1, \dots, \mathbf{y}_p\}$,

Mapping: $f : X \mapsto Y$, Training dataset: $Z = \{(\mathbf{x}_1, f(\mathbf{x}_1)), \dots, (\mathbf{x}_p, f(\mathbf{x}_p))\}$

Domain dependent input dataset: $X_D = \{\mathbf{x} | \mathbf{x} \in D \cap X\}$

Domain dependent training dataset: $Z_D = \{(\mathbf{x}, f(\mathbf{x})) | \mathbf{x} \in D \cap X\}$

Tree construction. A linear regression tree is constructed by Algorithm 1.

```

procedure Regression ( $Z$ )
begin
  Perform linier regression on  $Z$ 
  If the linear regression error on  $Z$ 
  is greater than given threshold then
     $Z$  is decomposed into  $Z_D$  and  $Z_{\overline{D}}$  ( $Z = Z_D + Z_{\overline{D}}$ );
  otherwise
    store the regression matrix to the node;
  return;
  Node ( $Z$ ) .up :=  $Z_D$ ; Node ( $Z$ ) .down :=  $Z_{\overline{D}}$ ;
  Regression ( $Z_D$ ); Regression ( $Z_{\overline{D}}$ );
end

```

Algorithm 1. Initial tree construction

This algorithm recursively performs linear regression and disjoint decomposition of domains until the regression error becomes smaller than given threshold. The domain decomposition rule employed here is to split the domain at the point on the most widely data-spreading axis into sub-domains having the same numbers.

As we discussed in Section 2, determining the splitting position is a difficult problem: if we insist on the optimality of the position, the tree construction time becomes considerably long, and if the input space is roughly split, the resulting regression becomes inaccurate and unreliable. For solving this problem, PaLM-tree [9] employs the following merge process so as to maximize the domain while keeping the error threshold. The merge procedure is described in Algorithm 2.

```

procedure Merge (Tree)
procedure test (domain1, domain2)
begin
  Perform regression on data in domain1 and domain2;
  If the error is smaller than given threshold then
    merge domain1 and domain2

```

```

otherwise return;
end
begin
  foreach adjacent domain(1), domain(2) ∈ LeafNodes
    test(domain1, domain2);
  end

```

Algorithm 2. Pseudo code for merging procedure of leaf nodes

By calling **Regression** followed by **Merge**, we can construct a liner regression tree.

Dimensionality reduction for regression. Through the tree construction, Algorithm 1 and 2 perform linear regressions using the training data in domains. If the data in a domain is insufficient compared with the dimensionality of the input vector, the regression result will be inaccurate and unreliable, because of the multi-collinearity problem. This problem can be solved by dimensionality reduction. PCR, PLS, CCA are the representative methods for dimensionality reduction. In this paper, we employ PLS (partial least squares), which reduces the input and output vectors so as to maximize the data covariation.

Error metric. Linear regression tree requires an error metric and threshold for terminating the domain decomposition. Suppose that $f_j(\mathbf{x}_i)$ is the j -th component of the classifier output for input \mathbf{x}_i and $g_j(\mathbf{x}_i)$ is that of regression result. Then, an error metric e in a domain D is defined as below.

$$e(D) = \sum_j \sum_{\mathbf{x}_i \in D} \left| g_j(\mathbf{x}_i) - f_j(\mathbf{x}_i) \right| \quad (1)$$

One may think that the error metric should be normalized by the size of D or the number of data in D . However, this metric is suitable for our task. If we normalize the error metric as $e/|D|$ or $e/|X_D|$, spiky error may be attenuated by other small errors and the decomposition may be terminated at shallow nodes.

For the classification, the absolute values of $f_j(\mathbf{x}_i)$ and $g_j(\mathbf{x}_i)$ are not essential but the ordering of values are important. For instance, the relation between j th element and k th element is $f_j(\mathbf{x}_i) \geq f_k(\mathbf{x}_i)$ and $g_j(\mathbf{x}_i) \geq g_k(\mathbf{x}_i)$, and $f_j(\mathbf{x}_i) < f_k(\mathbf{x}_i)$ and $g_j(\mathbf{x}_i) < g_k(\mathbf{x}_i)$ are consistent. Otherwise, the domain should be decomposed or two domains should not be merged, no matter how small $e(D)$ is.

Classification. Node n in a linear regression tree consists of 1) splitting component index $i(n)$, 2) splitting value $v(n)$, 3) pointers to offsprings (up(n), down(n)), 4) regression coefficient matrix $B_l(n)$. The $p \times q$ matrix $B_l(n)$ is stored only in a leaf node. For an input vector $\mathbf{x} = (x_1, \dots, x_p)^T$, the output $\mathbf{y} = (y_1, \dots, y_q)^T$ is computed by $\text{Traverse}(\text{root}, \mathbf{x})$ in Algorithm 3, where root represents the root node of the linear regression tree.

```

procedure Traverse( $n, \mathbf{x}$ )
begin
  if LeafNode( $n$ ) then return  $\mathbf{y} = B_1(n) \mathbf{x}$ ;
  else
    begin
      if  $x_{i(n)} < v(n)$  then
        return Traverse(down( $n$ ),  $\mathbf{x}$ );
      else
        return Traverse(up( $n$ ),  $\mathbf{x}$ );
      end
    end
  end

```

Algorithm 3. Traverse and output

From the output $\mathbf{y} = (y_1, \dots, y_q)^T$ of **Traverse**(), the classification result ω_k is obtained as,

$$k = \arg \max_{i=1, \dots, q} y_i. \quad (2)$$

3.1 Data Generation for Classifier Molding

As we described above, linear regression tree is too flexible, and hence, it has poor generalization power. However, the original classifier has some generalization power, which guides the linear regression learning so as to avoid over fitting to the training data. For bringing out the true generalization power of the original classifier, we have to provide large amount of data to the original classifier. While providing the data, the behavior of the classifier is learnt by linear regression tree.

Basically, the data generation is done as training-data multiplication with fluctuations. This data multiplication is simple and does not add new information. However, the trained classifier by generated data performs better than the classifier trained by original data [11]. In general, the fluctuation is done by adding Gaussian noise to the original data. The only thing we have to consider is in which space the fluctuation should be added.

In our case, IJP-OCR, we already know the font pattern of each character. This enables us to generate instances by adding fluctuations to the dot positions and sizes.

Our intention of training data multiplication is for training linear regression tree, but this is also effective for the original classifier. So, we use the generated data for both. For training the classifier, class labels have to be associated but no labels have to be associated for regression tree learning.

4 Classification Based Character Segmentation

After molding the classifier, we can get an accelerated imitation of a classifier. In our case, a fast character classifier is obtained. When the characters are well separated and correctly-segmented characters are provided for classification, no additional processing is required.

However, in the case of IIJP-OCR, there are many chances that bottom-up character segmentation fails, e.g., dot-position fluctuation, background noise (ex. cardboard spots), and so on. Binarization and labeling are the standard method of bottom-up character segmentation. Since IIJP character patterns consist of dots, dilation may be required to bridge unwanted gaps. However, this process can connect adjacent characters.

For solving this, we propose a top-down segmentation method named Classification based Character Segmentation (CCS), which finds the image segment sequence having maximum classification score. Since CCS classifies multiple image segment sequences, it requires a fast classifier. Fortunately, since we already have “the molded classifier”, we can realize a segmentation-robust character recognition system based on CCS.

CCS algorithm is based on the A* algorithm [12] that aggressively prunes off non-optimal image segment sequences. A* is an extension of Dijkstra’s algorithm [13], which is a graph search algorithm that solves the single-source shortest path problem.

4.1 CCS Problem

We assume that the vertical position of a character is already known. This assumption is valid for IIJP, because the height of IIJP head is fixed and known. Under this assumption, character center can be denoted by x position. Let $S(a)$ be the classification score at position a . CCS searches the character position sequence u_1, \dots, u_n between a_s to a_e in the input image that maximizes the total score $f(u_1, \dots, u_n) = S(u_1) + \dots + S(u_n)$. The maximum number of characters is n , and the positions have to be in the interval $[a_s, a_e]$. If one of these conditions is not satisfied, the search stops.

4.2 CCS Solution

Linear search of this problem has to compute $S(a)$ at all points within $[a_s, a_e]$. Since $S(a)$ computation requires character classification, the linear search requires too many character classifications, which slows down the speed. For solving this problem, we employ the framework of A* algorithm.

We assume that if a character presents at a , then $S(a) > \theta$. Also, the horizontal character interval is Δ . After the search, it will be able to get a maximum score positions $L = \{L_1, \dots, L_n\}$. Let’s suppose that K is distance-plus-cost heuristic value. It determines the order in which the search positions in the image. The path-cost function denoted $t^*(a)$, which is the cost from the a_s to the current position a , and $r^*(a)$ is an admissible “heuristic estimate” of the distance to the a_e . Under these assumptions, the algorithm can be described as below.

procedure CCS()

begin

$i := 1$; $u_0 = a_s$; $r^*(u_0) := n\theta$; $t^*(u_0) := 0$; $K = 0$;

Find the starting point u_1 within the interval

```

[ $a_s, a_s + \Delta$ ] that maximizes  $S(a)$ ;
Push (1,  $u_1$ ) into Stack;
while (!Empty(Stack))
begin
  Pop ( $i, u_i$ ) from Stack;
   $t^*(u_i) := t^*(u_{i-1}) + S(u_i)$ ;  $r^*(u_i) := (n-i)\theta$ ;
  if ( $r^*(u_i) + t^*(u_i) > K$ ) then {Pruning}
     $K := r^*(u_i) + t^*(u_i)$ ;  $L_i = u_i$ ;
  else if (( $i < n$ ) or ( $u_i \leq a_e$ )) then {Termination test}
    for ( $a = u_i, j = 0; a \leq u_i + \Delta; a++$ )
      if  $S(a) > \theta$  then  $c_j = a; j++$ ;
    Sort  $S(a)$  in descending order;
    Push ( $i+1, c_j$ ) into Stack in this order;
  end
return  $L$ ;
end

```

Algorithm 4. CCS algorithm

This algorithm performs the best first search and after reaching the maximum number of characters or a_e , it will start the backtracking to find better character positions.

We can issue an interrupt to terminate this algorithm, because IJJP-OCR has to output the read characters within a limited time. Even in this case, our algorithm may find better result compared with the binarization and labeling result. This is because our algorithm is basically the best first search. The behavior of this algorithm is illustrated in Fig. 1.

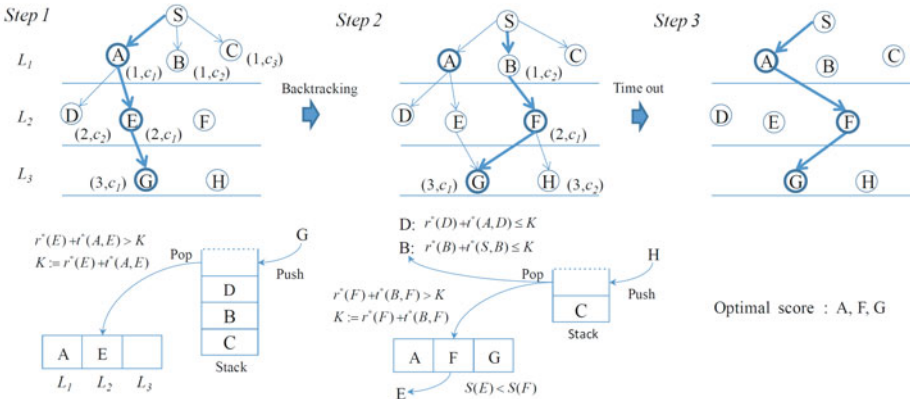


Fig. 1. Procedure of Classification based Character Segmentation. Step 1 shows procedure of A^* as usual. The procedure of Backtracking shown in Step 2, and Step 3 shows process of time-out that stops the search process and detects the optimal positions at this time.

5 Experiments

In the experiments, we first compare the performance of classifiers to select original classifiers. Then, by using selected classifier, we performed classifier molding for generating an accelerated classifier, whose accuracy and speed are examined. Finally, we compare the bottom-up and top-down character segmentation results.

We conducted all experiments on Windows-XP desktop PC with Intel Core2Duo 2.4 GHz CPU, 2 GB memory. Test images are 10000 real images. An example image is shown in Fig. 2. The training data are the mixture of real and generated images (16×16). The number of data for training original classifier is 512 for each character. Some examples are shown in Fig. 3. The character classes are 36 consisting of '0'-'9' and alphabet 'A'-'Z'. Further, we prepared three kinds of generated data sets with different standard deviations. We call each training data sets 1σ , 2σ , and 3σ , where σ represents a unit deviation.

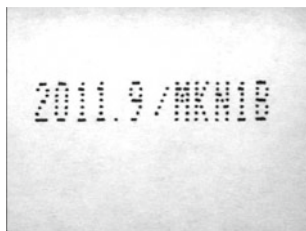


Fig. 2. An example of input image

Real image:



Generated image:




(1σ)  , (2σ)  , (3σ) 

Fig. 3. Example of training pattern

5.1 Classifier Comparison

We first compared the accuracy among four classifiers: NNM (Nearest Neighbor Method with single representative pattern per class), SVM, MSM (Multiple Similarity Method) [2], CSM. Table 1 shows the comparison of the recognition rate by each method using the real and generated images. NNM is the conventional method in commercially available IIJP-OCR. Note that, the SVM employed here is a linear SVM.

Table 1. Recognition rate by real image and generated image

Method	NNM	SVM	MSM	CSM
Real image (%)	97.5	94.23	95.47	99.98
1σ (%)	99.0	99.95	98.6	100.0
2σ (%)	91.68	99.95	99.22	100.0
3σ (%)	87.43	99.96	99.47	100.0

Table 1 shows that CSM is the most accurate method among them. Moreover, the accuracy of every classifier is improved by fluctuation of training pattern

becomes larger. This verifies that the effectiveness of using the generated images for training. For these reasons, we selected a CSM which trained with data set 3σ for classifier molding.

5.2 Classifier Molding

We performed classifier molding of the CSM, using input-output relationship of CSM for data set 3σ . Since this data set is generated by adding fluctuations to the font patterns, we can generate infinite number of data. The input is 256-dimensional vector, and output is 36-dimensional vector representing the similarities of the input to 36 classes. That regard $R^{256} \mapsto R^{36}$ nonlinear mapping problem.

First we examined the relationship between the number of data for classifier molding and the recognition rate of linear regression. Fig. 4 shows the resulted graph. From this graph, we confirmed that after providing 200 data per class, we get 100% recognition rate for all classes. This means the performance of CSM in Table 1 is completely imitated.

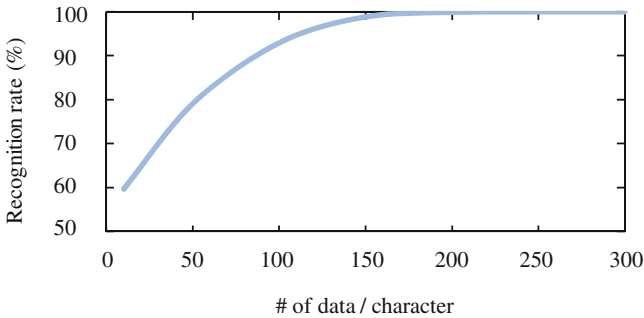


Fig. 4. The relation between the number of training data and recognition rate

Next, we measured the elapsed time for recognition per character. Table 2 shows the resulted mean time. We confirmed that the imitated classifier denoted by MLD is 1500 times faster than the original classifier CSM. It can be said achieve that accelerate the original classifier without dropping the recognition rate.

Table 2. Mean recognition time per character

Method	NNM	SVM	CSM	MLD
Time (msec)	9.8	15.2	15.0	0.01

5.3 Classification Based Character Segmentation

The above-mentioned experiments are the evaluations of classifier itself without segmentation errors. In this section, we show the result of character recognition

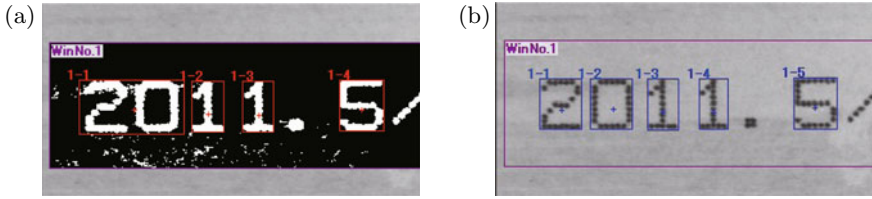


Fig. 5. (a) Segmentation error: ‘2’ is merged with ‘0’ by background noise. (b) Improvement by Classification based Character Segmentation.

ratio including character segmentation based on CCS. The test image is 512 gray images that cannot be correctly segmented by the binarization and labeling approach because of the touching characters as shown in Fig. 5 (a).

Based on our experience we set the number of extraction characters $n = 10$, similarity threshold $\theta = 0.4$, and width of character $\Delta = 40$. In this case, All test images are correctly segmented, and all characters are classified correctly. Fig. 5 (b) shows an example of successful segmentation. The mean processing time including segmentation and classification for an image is 20 ms (10 ms for segmentation and 10 ms for character classification). Compared with the ordinary IJJP-OCR, character extraction consumes 10 ms, classification by NNM consumes 98 ms, and the total elapsed time is 108 ms. This means our OCR system is 5 times faster, more accurate, and robust against segmentation errors than commercially available IJJP-OCRs.

6 Conclusion

This paper presents a method imitating arbitrary classifier by a linear regression tree that can be used as an accelerated classifier of the original classifier for improving the accuracy and speed of IJJP-OCR. Based on the accelerated classifier, we also present Classification based Character Segmentation (CCS) for avoiding character segmentation errors. In the experiments, we examined the accuracy of four classifiers and confirmed that CSM performs the best. By using the selected CSM, we examined the relationship between the recognition ratio and the number of training data for classifier molding, and confirmed that 200 training data per class are enough for imitating IJJP-OCR. The imitated classifier is 1500 times faster while keeping the accuracy. Also, we tested CCS and confirmed that 100% of the images are correctly segmented and classified, where all test images cannot be segmented by binarization and labeling approach.

This framework, classifier molding, can have wide application domains, however, it requires training data generation method. In our case, IJJP-OCR, we can get the original font patterns that can be used for data generation. If an application can have a data generation method or training data multiplication method, our framework, classifier molding can be applied, which accelerates the original classifier.

Future works involve training data multiplication method suitable for classifier molding, and further investigation of error metrics of linear regression tree suitable for classifier molding.

References

1. Quinlan, J.R.: Learning with continuous classes. In: Proceedings of 5th Australian Joint Conference on Artificial Intelligence, pp. 343–348. World Scientific Pub. Co. Inc., Tasmania (1992)
2. Ijima, T.: Pattern Recognition Theory. Morikita Shuppan, Japan (1989)
3. Vapnik, V.: The nature of statistical learning theory. Springer, Heidelberg (1995)
4. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Chapman & Hall, New York (1984)
5. Karalic, A.: Linear regression in regression tree leaves. In: Proceedings of ISSEK 1992 (International School for Synthesis of Expert Knowledge) Workshop, Bled Slovenia (1992)
6. Alexander, W.P., Grimshaw, S.D.: Treed regression. *Journal of Computational and Graphical Statistics* (5), 156–175 (1996)
7. Chaudhuri, P., Huang, M.-C., Loh, W.-Y., Yao, R.: Piecewise-polynomial regression trees. *Statistica Sinica* 4, 143–167 (1994)
8. Dobra, A., Gehrke, J.E.: SECRET: A Scalable Linear Regression Tree Algorithm. In: Proceeding of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton Alberta Canada (2002)
9. Nakamura, T., Kato, T., Wada, T.: A Novel NonLinear Mapping Algorithm (PaLM-Tree). *Journal of the Robotics Society of Japan* 23(6), 732–742 (2005)
10. Horowitz, S.L., Pavlidis, T.: Picture segmentation by a tree traversal algorithm. *Journal of The Association for Computing Machinery* 23(2), 386–388 (1976)
11. Murase, H.: Synthesized-based learning for image recognition. Technical report of IEICE 104(291), 41–48 (2005)
12. Hart, P.E., Nilsson, N.J., Raphael, B.: A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics* SSC-4(2), 100–107 (1968)
13. Dijkstra, E.W.: A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik* 1, 269–271 (1959)

Recognizing Continuous Grammatical Marker Facial Gestures in Sign Language Video

Tan Dat Nguyen¹ and Surendra Ranganath²

¹ Dept. of Electrical & Computer Engineering
National University of Singapore, Singapore 117576

`ntdat@nus.edu.sg`

² Indian Institute of Technology – Gandhinagar, India 382424
`surendra@iitgn.ac.in`

Abstract. In American Sign Language (ASL) the structure of signed sentences is conveyed by grammatical markers which are represented by facial feature movements and head motions. Without recovering grammatical markers, a sign language recognition system cannot fully reconstruct a signed sentence. However, this problem has been largely neglected in the literature. In this paper, we propose to use a 2-layer Conditional Random Field model for recognizing continuously signed grammatical markers in ASL. This recognition requires identifying both facial feature movements and head motions while dealing with uncertainty introduced by movement epenthesis and other effects. We used videos of the signers' faces, recorded while they signed simple sentences containing multiple grammatical markers. In our experiments, the proposed classifier yielded a precision rate of 93.76% and a recall rate of 85.54%.

1 Introduction

Interpreting sign language not only requires recognition of hand gestures/signs, but also other non-manual signs. As pointed out in [1], non-manual signs convey important grammatical information. Without these grammatical markers, the same sequence of hand gestures can be interpreted differently. For example, with the hand signs for BOOK and WHERE, a couple of sentences can be framed as

- $[BOOK]_{TP} [WHERE]_{WH} \rightarrow$ Where is the book?
- $[BOOK]_{TP} [WHERE]_{RH} \rightarrow$ I know where the book is!

In the notation of the above example, the left hand side of the arrows represent signs in American Sign Language (ASL). The subscripts TP, WH and RH on the words BOOK and WHERE indicate grammatical markers conveyed by facial feature movements and head motions. The facial gesture for Topic (TP) is used to convey that BOOK is the topic of the sentence. The word WHERE accompanied by a WH facial gesture, signals a “where?”. The hand sign for WHERE made concurrently with the facial gesture for RH indicates the rhetorical nature of the second sentence.

Thus, recognition of non-manual signs is required for building a complete sign language understanding system. However, review [2] of sign language recognition indicates that the dominant interest in sign language recognition has been in hand gesture recognition. Non-manual sign recognition has only recently started to receive attention [3] [4].

Previous works on recognizing facial expressions were reviewed in [5] and [6]. These surveys showed that many works focused on recognizing the six isolated universal expressions (Anger, Disgust, Fear, Happiness, Sadness, and Surprise) with minimal head motion. The latter simplification of the problem makes these methods inapplicable for recognizing facial gestures in sign language, where facial expressions are defined concurrently with head motion to define grammatical markers. There are also many works on analyzing head pose and head motion [7]. However, there are few works in the literature that address recognizing facial expressions coupled with concurrent head motion.

Black and Yacoob's work [8] is a pioneering work in recognizing continuous facial expressions with head motion based on features from dense optical flow and rule-based discriminative models. They obtained an average recognition rate of 88% and 73% on laboratory data and data from TV programs, respectively. De la Torre et al. [9] proposed to detect rare facial gestures made during an interview based on Personalized Active Appearance Model [10]. However, quantitative assessment of the detection was not reported. Cohen et al. [11] used a piecewise 3D wire frame model-based approach for tracking 16 facial features and estimated their 3D motions. These were used in a multi-level HMM scheme for classifying the six universal expressions and the neutral expression in video sequences containing multiple expressions. They reported 82.46% and 58.63% accuracy for person dependent and person independent tests, respectively, on their database of 5 persons.

As generative models, HMMs suffer from two weaknesses: the statistical independence assumption of observations and the difficulty in modeling their complicated underlying distributions. On the other hand, Conditional Random Fields (CRF) proposed by Lafferty et al. [12] is a discriminative model which avoids these weaknesses. Kanaujia and Metaxas [13] used the CRF to recognize the six universal expressions and obtained promising results. Quattoni et al. [14] proposed Hidden-state CRF (HCRF) models and obtained an accuracy of 85.25% for recognizing head shakes and head nods. Chang et al. [15] proposed a modified HCRF called Partially-Observed HCRF (PO-HCRF). The PO-HCRF achieved an accuracy of 80.1% with 9.18% false alarm rate for recognizing the six "continuous" universal facial expressions in simulated sequences created by concatenating sequences of isolated expressions. Neidle et al. [4] proposed to detect the presence of *WH* and *NEG* grammatical markers in ASL signed sentences. An ASM-based tracking scheme proposed was used to track face and facial feature movements, and provide head pose (pitch, yaw, and tilt) in each frame. Each video frame was classified as either *WH* or *not-WH*, and a video sequence was labeled based on majority voting of frames. A multiple-SVM classifier was used

to label each frame. The recognition accuracies were 100% and 95% for *WH* and *NEG*, respectively.

In this paper, we consider recognizing continuous facial gestures in sign language, particularly grammatical markers in ASL. The six grammatical markers considered in this paper are summarized in Table 1 in terms of eye, eyebrow, and head movements. We propose to use a layered Conditional Random Field (CRF) model [12] for this purpose. The classifier includes two CRF layers, the first layer to model head motions and the second to model grammatical markers. The separate head motion layer helps to reduce the ambiguity in recognizing grammatical markers in the second layer. For each video sequence, probabilities of different head motions are evaluated by the first layer, and these are input to the second layer together with other features for labeling the grammatical marker for each frame. Manually annotated labels of head motions and grammatical markers were used for training the classifier and assessing performance. The classifier yielded precision and recall rates of 95.24% and 85.54%, respectively.

2 Recognizing Continuous Facial Gestures in Sign Language

2.1 Challenges

Facial gestures in ASL are identified from head motion and facial feature movement. In this paper we consider recognition of six grammatical markers listed and described in Table 1, through their head gestures comprising, eye, eyebrow and head movements. In previous work [16], we have considered recognition of isolated facial gestures. Here, we extend our work to recognition of continuous facial gestures as would occur in sign language discourse, and consider four types of facial gesture sequences (Table 2) composed of these grammatical markers. Examples of these facial gesture chains are shown in Table 3.

There are several aspects to the continuous facial gesture recognition problem which make it challenging, more so than isolated recognition. *Movement epenthesis* is the extra motion required by the head (and facial features), due to physical constraints, to transit from the end of the previous gesture to the beginning of

Table 1. Simplified description of the six ASL grammatical markers (Exp.) considered: *Assertion(AS)*, *Negation(NEG)*, *Rhetorical(RH)*, *Topic(TP)*, *Wh question(WH)*, and *Yes/No question(YN)*. Nil denotes unspecified facial feature movements.

Exp.	Brow	Eye	Head
<i>AS</i>	Raise	Nil	Nod
<i>NEG</i>	Knit	Nil	Shake
<i>RH</i>	Raise	Widen	Tilt(left/right)
<i>TP</i>	Raise	Widen	Move upward
<i>WH</i>	Knit	Squint	Move Forward
<i>YN</i>	Raise	Widen	Move Forward

Table 2. Types of grammatical marker sequences considered

Sequence	English sentence	ASL signs
<i>TP AS</i>	I really want the book!	[BOOK] _{TP} [WANT] _{AS}
<i>TP NEG</i>	I don't want the book.	[BOOK] _{TP} [WANT] _{NEG}
<i>TP RH AS</i>	I know where the game is! It's in Singapore.	[GAME] _{TP} [WHERE] _{RH} [SINGAPORE] _{AS}
<i>TP WH YN</i>	Where is the game? Is it in New York?	[GAME] _{TP} [WHERE] _{WH} [NEW YORK] _{YN}



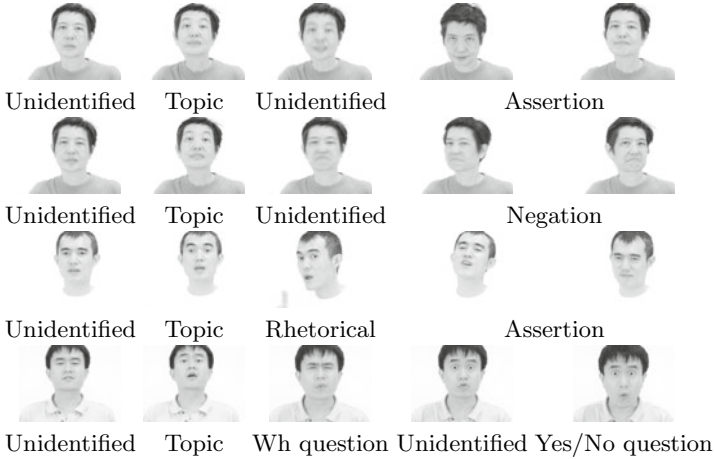
Fig. 1. When the *Rhetorical* gesture is performed after a *Topic* gesture, the head will move from backward position to neutral position before tilting forward (movement epenthesis) while the brow still held raised

the next; this is difficult to model due to its variability. *Coarticulation* refers to the appearance of a head gesture being influenced by adjacent gestures. There can also be asynchronization between head motion and facial feature movement. Movement epenthesis between grammatical markers is shown in Fig. 1. Table 3 shows examples of grammatical marker chains; any facial gesture video frame that does not contain one of the six grammatical marker classes is labeled as Unidentified. This is a generic class which includes gestures between two grammatical markers, and also the neutral expression, which is usually present at the beginning of a sequence.

Visually, the beginning and ending of an expression can be considered to coincide with the beginning and ending of the head motion corresponding to that expression. However, while signing, movements of facial features like brows and eyes are independent and may evolve asynchronously with the head motion. This asynchronization adds to the uncertainty in identifying a facial gesture by using a combination of features from head motions and facial feature movements. An effective strategy to deal with this problem is to use multi-channel frameworks [17], where the classifier learns the correlations between the channels through supervised training.

Movement epenthesis between grammatical markers also introduces additional variability. This is manifested through the head tending to move back to the neutral position before comfortably starting the next motion. Besides, if expressions have similar eye/brow movements, some subjects tend to hold the state established at one expression into the next expression, while others do not. This phenomenon will alter the temporal patterns of eye/brow movements and affect algorithm performance. The movements of the eyes and brows can be further affected by factors that are not related to facial gestures of interest: natural eye

Table 3. Examples of four types of grammatical marker chains. The neutral expression shown in the first frame is considered to be an unidentified expression. An unidentified facial gesture can also be present between any two grammatical markers and can vary greatly depending on nearby grammatical markers.



blinks, hand signs for adjectives such as HUNGRY or FAST involving added facial expressions.

Moreover, unidentified gestures between facial gestures of interest are highly varied due to combinations of movement epenthesis and other effects. Thus, it will be ineffective to model the sequences using generative models like HMMs. A discriminative model may be more suited for this scenario, and we propose to use a 2-layer CRF model to handle head motion and facial expression towards recognizing continuous grammatical markers. The use of a 2-layer model is also motivated by the experimental data that we gathered, which showed that in spite of movement epentheses, head motions are more consistent than corresponding facial feature movements.

2.2 Layered Conditional Random Field Model

The CRF is a discriminative probabilistic model proposed by Lafferty et al. [12] which can be trained to assign a sequence of predefined labels to a sequence of observations. Its evaluation function is composed of weighted potential functions which can utilize not only features extracted from the observations but also their interactions and temporal dependencies. In the linear-chain model, the probability of a label sequence \mathbf{y} given an observation sequence \mathbf{x} is computed as:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp \sum_{t=1}^T \left(\sum_{i=1}^N \lambda_i f_i(y_t, \mathbf{x}) + \sum_{j=1}^M \mu_j g_j(y_t, y_{t-1}, \mathbf{x}) \right) \quad (1)$$

Table 4. Head labels used to train the CRF at the first layer

No.	Label	Meaning
1	Neutral (<i>Neu</i>)	Head at normal position
2	Forward (<i>Fw</i>)	Head moves forward
3	Back from Forward (<i>BfF</i>)	Head moves from forward position to neutral position
4	Backward (<i>Bw</i>)	Head moves backward
5	Back from Backward (<i>BfB</i>)	Head moves from backward position to neutral position
6	Turn left (<i>TL</i>)	Head turns left, usually a part of head shake
7	Back from Turn left (<i>BfTL</i>)	Head pose changes from leftward to frontal
8	Turn right (<i>TR</i>)	Head turns right, usually a part of head shake
9	Back from Turn right (<i>BfTR</i>)	Head pose changes from rightward to frontal
10	Move down (<i>MD</i>)	Head moves down, usually a part of head nod
11	Back from Move down (<i>BfMD</i>)	Head pose changes from downward to frontal, usually a part of head nod
12	Still	Head is kept still
13	Forward left (<i>FL</i>)	Head moves forward and slightly turns left
14	Back from Forward left (<i>BfFL</i>)	Head pose changes from leftward to frontal and head moves from forward to neutral position
15	Forward right (<i>FR</i>)	Head moves forward and slightly turns right
16	Back from Forward right (<i>BfFR</i>)	Head pose changes from rightward to frontal and head moves from forward to neutral position

where f_i and g_j are potential functions that evaluate the interaction and temporal dependencies among features, respectively. λ_i and μ_j are weights estimated from training data, and $Z(\mathbf{x})$ is a normalization factor.

It was shown [12] that the right hand side of Eq. 1 is a convex function parameterized by λ_i and μ_j , whose global optimum can be obtained by using iterative scaling algorithms or gradient-based methods.

CRFs, which avoid the assumption of statistical independence of observations, have shown better performance than HMMs in many applications [12] [14]. We use a layered model of the chain CRF (Fig. 2) to recognize continuous facial gestures in ASL. The probabilities of head motion labels are evaluated by a CRF in the first layer. These probabilities are passed to the second layer where other facial feature channels are also integrated. The second layer CRF is trained on these integrated features, to provide grammatical marker labels for frames in the test video sequences.

Our observations show that the transition from one type of head motion to another mainly include movement epenthesis. Thus we choose to model movement epentheses explicitly, together with meaningful head motions. Currently, we have used 16 labels of head motions (both meaningful head motion and their movement epentheses) as described in Table 4 for all combinations of head motions which occur in conjunction with the six grammatical markers of interest.

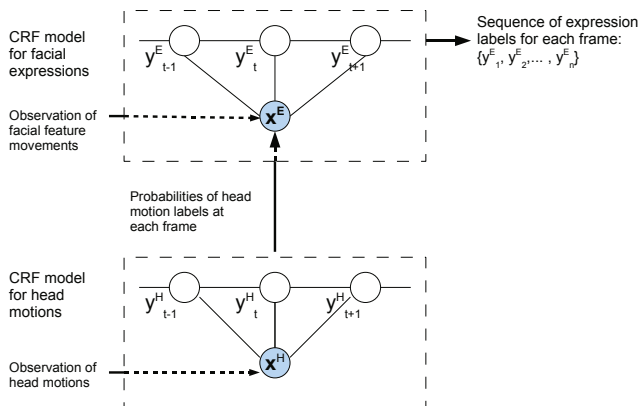


Fig. 3. Feature points of interest

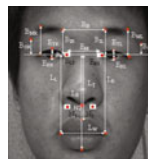


Fig. 4. Distance features used

Fig. 2. Layered CRF for recognizing continuously signed grammatical markers in sign language

In manually annotating the frames, besides the head motion label, each video frame in the data set is also labeled with one of seven facial gestures: *AS*, *NEG*, *RH*, *TP*, *RH*, *WH*, *YN*, and *Und*. The label *Und* is assigned to frames with unidentified expressions.

As shown in Table 4, head motions with labels such as “Back from X” are defined to explicitly model movement epentheses. Exceptional cases are labels 7, 9, and 11 which are constituents of multi-part head motions: head shake and head nod. The *Neutral* label appears mostly at the beginning of the video sequences. During facial gestures, the head does move past the neutral position but does not stop. The frames in which the head is temporarily at the neutral position is also annotated with the *Neutral* label. The label *Still* plays an important role in segmenting meaningful head motions and their movement epentheses (Back from X) because there is usually a short pause (or even long pause) between the meaningful head motion and its “Back from” movement.

Motion of the head and facial features are obtained from the tracked feature points (shown in Fig. 3) using an enhanced version of the robust tracking algorithm developed by the authors [16]. The feature points are placed at both rigid and non-rigid facial locations, and distances between them are extracted and used for recognition. These distances (shown in Fig. 4) are, (a) five eyebrow parameters: *Left inner brow height* (B_{IL}), *Right inner brow height* (B_{IR}), *Left middle brow height* (B_{ML}), *Right middle brow height* (B_{MR}), *Distance between brows* (B_B); and (b) two eye parameters: *Left eye height* (summation of E_{BL} and E_{TL}), *Right eye height* (summation of E_{BR} and E_{TR}). A reference line is defined as the line passing through the two inner eye corners, and the height parameters are the perpendicular distances of the feature points from this line. All distance parameters are normalized with respect to their corresponding values in the first frame to remove scaling effects across video sequences.



Fig. 5. Frames in a test sequence containing the facial gesture chain $TP RH AS$. The frame index is shown below each image. Blue dots at facial features of interest are our tracking results.

To recognize head motions, tracks of non-deformable facial feature locations, namely, the two inner eye corners (E_{L3} , E_{R3}) and the middle of the nose (N_2), are used to define three features; S_M (the area of the triangle formed by the above three locations in each frame), and C_{Mx} , C_{My} (components of the 2D motion vector¹ C_M of the center of gravity of the triangle). S_M and C_M are normalized by the distance E_{M0} between the two inner eye corners in the first frame: $C_{Mt}^n = \frac{C_{Mt}}{E_{M0}}$ and $S_{Mt}^n = \frac{S_{Mt}}{E_{M0}^2}$. These three features form the feature vector (at each frame) for the first CRF layer to evaluate probabilities of different head motions. The feature vector (at each frame) of the second CRF layer for recognizing continuous grammatical markers thus has 23 elements: 16 probabilities of head motions and 7 distance ratios computed from the eyes and brows' tracked features.

3 Experiments and Results

Videos of natural sign language facial gestures of interest were recorded by providing deaf signers (from the Deaf and Hard-of-Hearing Foundation of Singapore) with appropriate signing scripts for sentences. Each English sentence in the script was signed in ASL with hand signs and corresponding facial gestures. These sentences were created or adapted from ASL resources (e.g. [1]). A subject signed each sentence ten times. As mentioned in Section 2, the data includes four types of grammatical marker chains described in Table 2.

All six grammatical markers listed in Table 1 are present in the data set together with the 16 types of head motion described in Table 4. For evaluating the feasibility of our proposed recognition method, data from three subjects was used for experiments. The data set included a total of 129 video sequences divided into 93 video sequences for training (an average of seven sequences per subject for each of the four grammatical marker chains) and 36 for testing (about 3 sequences per subject per chain). Each video frame was manually transcribed to have two labels, one for head motion, and the other for grammatical marker, both identified based on visual observation and the signing script. The training set was used to train both CRF layers of the model: head motion layer and grammatical marker layer.

Recognition accuracy for grammatical markers was measured by two methods: frame based and label-aligned. In the frame-based method, the label assigned

¹ Motion vector $\mathbf{v}_{t+1} = (x_{t+1}, y_{t+1}) - (x_t, y_t)$.

for each frame is compared with the corresponding human annotated label. In the label-aligned method, the frame labels of each sequence are reduced such that consecutive frames with the same label are replaced by a single label. The two reduced sequences of labels are aligned using the Needleman-Wunsch algorithm [18]. The number of matches, insertions, deletions, and changed labels are then obtained. Insertions are labels output by the classifier, which do not appear in the corresponding annotated data. Deletions are labels which are not recognized by the classifier while they appear in the annotated data.

An experiment was conducted to evaluate the performance of the proposed model. The first CRF layer for head motion was trained first. The head motion probabilities output by this trained CRF was used as a part of the training vector for the CRF at the second layer. The two CRF layers were trained using the scaled conjugate gradient algorithm with the CRF Toolbox [19].

Frames from a video sequence in the test set are shown in Fig. 5, where the sequence of facial gestures corresponds to *TP RH AS*. Fig. 6 shows the probability output of the first layer for the 16 head motion labels described in Table 4. As mentioned in Section 2, the head tends to move past the neutral position before starting a new motion. In the last 10 frames in Fig. 6, there is confusion due to ambiguous head motions at the end of the signed sentence. Fig. 7 shows the probability for the grammatical markers output by the 2-layer CRF classifier. Seven probabilities including six for grammatical markers and one for unidentified expression are obtained at each frame. Fig. 7 shows that the second CRF layer, which is trained with output from the first layer, can tolerate the ambiguity of head motions in recognizing continuous grammatical markers.

The average frame-based grammatical marker recognition rate using the complete 2-layer CRF model was 80.82%. The corresponding confusion matrix is shown in Table 5 which shows that most of the confusions are between any grammatical marker and the unidentified expression. Particularly, frame-based label confusions occur at the boundary between facial gestures where ambiguous head motions and asynchronous movements of facial features are present. This makes even manual annotation of consecutive frames into different facial gestures difficult.

The label-aligned method of computing accuracy reveals more about the capability of the layered CRF for recognizing continuous grammatical markers by discounting unavoidable confusions during transitions between facial gestures. Table 5 can be augmented with insertion and deletion entries to obtain the extended confusion matrix \mathbf{C} from which precision and recall rates are computed as: $Precision = \frac{Match}{Match+Change+Insert}$ and $Recall = \frac{Match}{Match+Change+Delete}$, where for marker i , Match rate = $\mathbf{C}(i, i)$, Change rate = $\sum_{j \notin \{i, Insert, Delete\}} \mathbf{C}(i, j)$, Insertion rate = $\mathbf{C}(i, Insert)$, Deletion rate = $\mathbf{C}(i, Delete)$, and $\mathbf{C}(i, j)$ is the value at row i and column j of the extended confusion matrix.

Label-aligned results were 93.76% for precision and 84.54% for recall. The extended confusion matrix for this evaluation is shown in Table. 6. The precision rate appears quite reasonable given the complexity of the problem. However,

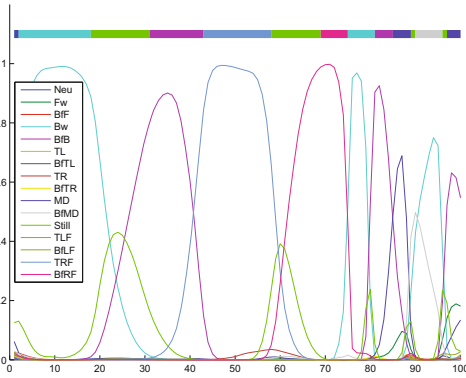


Fig. 6. The probability outputs of the first layer CRF trained to recognize 16 types of head motion. The color bar at the top is the human annotated head motion label for this video sequence. The curve and bar with the same color are associated with the same head motion. Labels for last 10 frames are ambiguous due to ambiguous head motions at the end of the signed sentence.

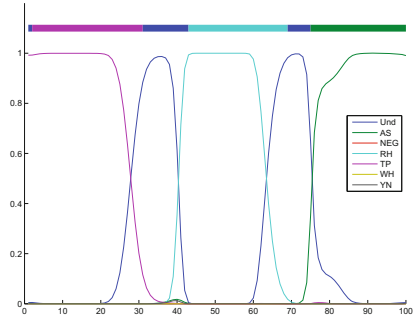


Fig. 7. The probabilities of the grammatical markers, output by the second CRF layer trained using head motion probability output (shown in Fig. 6) from the first layer

the lower recall rate hints that the layered CRF is less sensitive to change of facial gestures in video sequences. This may be improved with more descriptive features for head motion and facial feature movements. As a comparison, the results obtained in this experiment were quite close to the results we obtained in another experiment where the head motion labels were assumed known (the human annotated labels) and were input to the second layer CRF (rather than using the first layer outputs). In this experiment, precision rate of 94.54% and recall rate of 90.78% were obtained for recognizing grammatical markers. Our recent results show that the layered-CRF model outperforms the linear chain CRF and the layered HMM models.

Table 5. Confusion matrix for labeling grammatical markers with the proposed model. The average frame-based recognition rate is 80.82%.

Und	AS	NEG	RH	TP	WH	YN
59.62	7.60	3.09	6.65	9.26	12.11	1.66
9.62	87.46	0	2.92	0	0	0
0.98	0	97.07	0	1.95	0	0
10.78	0	0	89.22	0	0	0
3.06	1.31	1.17	3.35	91.1079	0	0
5.61	9.35	0	0	0	84.58	0.46
27.84	10.31	0	0	0	5.16	56.70

Table 6. Extended confusion matrix for label-based facial gesture recognition result (%) using 2-layer CRF

	UN	AS	NEG	RH	TP	WH	YN	Insert	Delete	Precision	Recall
UN	68.97	0.00	0.00	0.00	0.00	0.00	0.00	3.45	27.59	95.24	71.43
AS	5.26	84.21	0.00	5.26	0.00	0.00	0.00	0.00	5.26	88.89	84.21
NEG	0.00	0.00	100	0.00	0.00	0.00	0.00	0.00	0.00	100	100
RH	0.00	0.00	0.00	100	0.00	0.00	0.00	0.00	0.00	100	100
TP	0.00	0.00	0.00	0.00	91.67	0.00	0.00	0.00	8.33	100	91.67
WH	0.00	11.11	0.00	0.00	0.00	88.89	0.00	0.00	0.00	88.89	88.89
YN	0.00	11.11	0.00	0.00	0.00	0.00	55.56	0.00	33.33	83.33	55.56
Average										93.76	84.54

4 Conclusion

In this paper, we addressed the problem of recognizing continuous facial gestures in sign language video. A 2-layer CRF was proposed for recognizing six common grammatical markers in ASL sentences. The first layer was trained for evaluating head motions and the second layer was trained for segmenting and recognizing facial gestures using the output from the first layer and measurements of facial feature movements. Data was collected using an experimental set up for capturing natural facial gestures without a forced “neutral” state between gestures. The performance of the complete 2-layer CRF model yielded precision rate of 93.76%, and recall rate of 85.54% for recognizing the six types of continuously signed grammatical markers. These encouraging results show that the proposed 2-layer model is a viable scheme for recognizing facial gestures in sign language. In the near future, we propose to enhance the robustness of the model by incorporating more descriptive features for identifying head motions. We will also conduct more evaluations and comparisons with other methods. Other non-manual signals will be considered for further development of the system.

Acknowledgement. This work is partially support by project grant NRF2007IDM-IDM002-069 on “Life Spaces” from the IDM Project Office, Media Development Authority of Singapore.

References

1. Baker, C., Cokely, D.: American Sign Language: A teacher’s Resource Text on Grammar and Culture. Clerc Books, Gallaudet University Press, Wasington D.C. (1980)
2. Ong, S., Ranganath, S.: Automatic Sign Language Analysis: A Survey and the Future Beyond Lexical Meaning. IEEE Transactions on Pattern Analysis and Machine Intelligence 27, 873–891 (2005)
3. Vogler, C., Goldenstein, S.: Facial movement analysis in ASL. Journal on Universal Access in the Information Society 6, 363–374 (2008)

4. Neidle, C., Nash, J., Michael, N., Metaxas, D.: A Method for Recognition of Grammatically Significant Head Movements and Facial Expressions, Developed Through Use of a Linguistically Annotated Video Corpus. In: Proceedings of the Language and Logic Workshop, Formal Approaches to Sign Languages, European Summer School in Logic, Language, and Information (ESSLLI 2009), Bordeaux, France (2009)
5. Pantic, M., Rothkrantz, L.J.: Automatic Analysis of Facial Expressions: The State of the Art. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 1424–1445 (2000)
6. Fasel, B., Luetttin, J.: Automatic facial expression analysis: a survey. *Pattern Recognition* 36, 259–275 (2003)
7. Murphy-Chutorian, E., Trivedi, M.M.: Head pose estimation in computer vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31 (2009)
8. Black, M., Yacoob, Y.: Recognizing facial expressions in image sequences using local parameterized models of image motion. *International Journal of Computer Vision* 25, 23–48 (1997)
9. la Torre, F.D., Campoy, J., Ambadar, Z., Cohn, J.F.: Temporal Segmentation of Facial Behavior. In: International Conference on Computer Vision (2007)
10. Matthews, I., Baker, S.: Active Appearance Models Revisited. *International Journal of Computer Vision* 60, 135–164 (2004)
11. Cohen, I., Sebe, N., Garg, A., Chen, L.S., Huang, T.S.: Facial expression recognition from video sequences: temporal and static modeling. *Computer Vision and Image Understanding* 91, 160–187 (2003); Special Issue on Face Recognition
12. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labelling sequence data. In: International Conference on Machine Learning (2001)
13. Kanaujia, A., Metaxas, D.: Recognizing Facial Expressions by Tracking Feature Shapes. In: International Conference on Pattern Recognition, Hong Kong, China (2006)
14. Quattoni, A., Wang, S.B., Morency, L.P., Collins, M., Darrell, T.: Hidden Conditional Random Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29, 1848–1852 (2007)
15. Chang, K.Y., Liu, T.L., Lai, S.H.: Learning partially-observed hidden conditional random fields for facial expression recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 533–540 (2009)
16. Nguyen, T.D., Ranganath, S.: Tracking facial features under occlusions and recognizing facial expressions in sign language. In: IEEE International Conference on Automatic Face & Gesture Recognition, Amsterdam, Netherlands, pp. 1–7 (2008)
17. Oliver, N., Horvitz, E., Garg, A.: Layered representations for learning and inferring office activity from multiple sensory channels. *Computer Vision and Image Understanding* 96, 163–180 (2004)
18. Needleman, S.B., Wunsch, C.D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology* 48, 443–453 (1970)
19. Schmidt, M., Swersky, K.: Conditional Random Field Toolbox for Matlab, <http://www.cs.ubc.ca/~murphyk/Software/CRF/crf.html>

Invariant Feature Set Generation with the Linear Manifold Self-organizing Map

Huicheng Zheng

School of Information Science and Technology, Sun Yat-sen University
Guangzhou, China
zhenghch@mail.sysu.edu.cn

Abstract. One of the most important challenges faced by computer vision is the almost unlimited possibilities of variation associated with the objects. It has been hypothesized that the brain represents image manifolds as manifolds of stable neural-activity patterns. In this paper, we explore the possibility of manifold representation with a set of topographically organized neurons with each representing a local linear manifold and capturing some local linear feature invariance. In particular, we propose to consider the local subspace learning at each neuron of the network from a Gaussian likelihood point of view. Robustness of the algorithm with respect to the learning rate issue is obtained by considering statistical efficiency. Compared to its predecessors, the proposed network is more adaptive and robust in learning globally nonlinear data manifolds, which is verified by experiments on handwritten digit image modeling.

1 Introduction

In computer vision as well as in many other artificial perception problems, one of the most challenging issues is variation of the observations. A given object can be projected on the retina as very different images due to distance, location, orientation, lighting, etc., not to mention that the object itself may undergo complex distortion. Images of objects under continuous variability are often considered to lie on some intrinsic low-dimensional manifolds [14]. Manifold learning approaches such as the locally linear embedding (LLE) [13] and the isometric feature mapping (ISOMAP) [15] have been proposed in the literature and attracted wide interest. In this paper, however, we are interested in models that provide some kind of abstraction of data classes, which are more practical in visual perception problems for their capacity of invariant representation, generalization to future data, and robustness to noise or over-fitting.

Previous research has demonstrated that distributions of many images, e.g. handwritten characters and faces, can be effectively modeled by low-dimensional linear subspaces [3,7], which are special manifolds in the input data space. A typical strategy is to train a subspace model from images of all classes. A subspace learned in this way is a representation of the common object, e.g. an average face. Any dimensions orthogonal to the subspace are regarded as noise and will

be ignored. Projections on this subspace will then reveal the essential information that characterizes the various classes. Another strategy is to model classes directly as subspaces. Indeed, important transformation groups can be automatically taken into account if the classes are modeled as linear subspaces [11]. In general, distributions of class data are nonlinear. It is therefore necessary for the model to be adaptive to nonlinear distributions, which can be defined as low-dimensional nonlinear manifolds embedded in the data space. It has been hypothesized that the brain represents image manifolds as manifolds of stable neural-activity patterns [14]. Recalling the elegant and amazing performance of the brain, it is interesting to realize manifold learning and feature invariance by following a biologically plausible strategy.

The self-organizing map (SOM) has been extensively implemented in tasks related to computer vision [5]. A SOM consists of a grid of processing units each containing a weight vector. The map learns a topographically organized low-dimensional representation of the input space, where nearby weight vectors are similar while dissimilar weight vectors are far from each other. Certain assumptions of the SOM theory seem to have biological counterparts, and the same principle might underlie the emergence of feature maps in the living brain [5]. To achieve Gabor-like feature invariance, subspace learning has been introduced into neurons of the SOM [6]. In the new model, named the adaptive-subspace self-organizing map (ASSOM), the single weight vectors at map units in the SOM are replaced by sets of basis vectors that span some linear subspaces. By setting filters to correspond to pattern subspaces, some transformation groups, such as translation, rotation, and scaling can be automatically incorporated [6].

However, the ASSOM may not be adequate in learning nonlinear manifolds embedded in the data space, since subspaces in the ASSOM must pass through the origin. To overcome this limit, a number of variants have been proposed, such as the adaptive-manifold self-organizing map (AMSOM) [9] and the principal components analysis self-organizing map (PCASOM) [10]. A common difficulty of these ASSOM-type networks is the confusion between local sub-models due to infinite extensions of local subspaces. Therefore, some variants with more localized subspace representation are proposed [11,18], which suggest to include distances to local mean vectors into the objective function. However, the weight of this component can only be determined beforehand and empirically [11,18]. In this paper, we propose to consider the objective function from a likelihood point of view, which leads to a model more adaptive to complex nonlinear manifolds embedded in the data space, such that shapes of local distributions can be automatically captured by variances in the principal directions. Robustness of the proposed algorithm with respect to the learning rate is obtained by considering statistical efficiency. To guarantee a local principal subspace solution, we further incorporate constraints related to the mean squared error.

The rest of this paper is organized as follows. Section 2 presents local linear manifold modeling at each neuron for invariant feature generation. The overall model for generating a topologically ordered invariant feature set is presented in Section 3. The capacity of the proposed method to learn nonlinear manifolds for

visual object representation is demonstrated in Section 4 through experiments on handwritten digit image modeling. Finally, Section 5 concludes this paper.

2 Local Linear Manifold Learning for Invariant Feature Representation at Each Neuron

A pattern that undergoes certain transformations can be thought to occupy a low-dimensional nonlinear manifold in the vector space, which is an invariant representation of that pattern class. When the transformations are restricted or only considered locally, we may equate local linear manifolds at neurons with pattern classes subject to certain linear transformations.

A common practice in recent study [1,18] for learning nonlinear manifolds with local linear models is to take into account the distances of training vectors to centers of the local models to avoid infinite extension along local linear manifolds. The reconstruction error at each neuron takes the following form:

$$e(\mathbf{x}, \mathcal{L}) = \|\tilde{\mathbf{x}}\|^2 + \alpha \|\hat{\mathbf{x}}\|^2 \quad (1)$$

where $\hat{\mathbf{x}}$ and $\tilde{\mathbf{x}}$ are the projection and the error of the input vector \mathbf{x} on the local linear manifold \mathcal{L} , respectively, $0 \leq \alpha \leq 1$ is a weight parameter that controls the localization of the linear manifold [18]. However, traditional methods, which aim to minimize the expectation $E[e(\mathbf{x}, \mathcal{L})]$, are ill-defined with respect to α and will converge to $\alpha = 0$, as shown in [1].

2.1 The Gaussian Likelihood Function

In this paper, we formulate the optimization problem from a maximum likelihood point of view. Let D be the dimension of the input vector, \mathcal{L} be a linear manifold represented by the local model, H be the dimension of \mathcal{L} , \mathbf{m} be the mean vector associated with \mathcal{L} , \mathbf{b}_h and σ_h^2 be, respectively, the h -th basis vector and the variance in the direction of \mathbf{b}_h for $h \in \{1, 2, \dots, H\}$, and σ_0^2 be the variance in the directions of noise (orthogonal to \mathcal{L}). The basis vectors \mathbf{b}_h are assumed to be orthonormal. If \mathbf{b}_h have been chosen as principal eigenvectors of the underlying distribution, then for an input vector \mathbf{x} , the Gaussian likelihood function can be defined as

$$p(\mathbf{x}|\mathbf{m}, \mathbf{b}_h, \sigma_h) = \frac{\exp\left(-\frac{1}{2}\left(\sum_{h=1}^H ((\mathbf{x} - \mathbf{m})^T \mathbf{b}_h)^2 / \sigma_h^2 + \tilde{\mathbf{x}}^T \tilde{\mathbf{x}} / \sigma_0^2\right)\right)}{(2\pi)^{D/2} \sigma_0^{D-H} \prod_{h=1}^H \sigma_h} \quad (2)$$

where $\tilde{\mathbf{x}}$ is the projection error vector defined by

$$\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}} = \mathbf{x} - \sum_{h=1}^H ((\mathbf{x} - \mathbf{m})^T \mathbf{b}_h) \mathbf{b}_h \quad (3)$$

We may also consider the log-likelihood function

$$L_o(\mathbf{x}) = \ln p(\mathbf{x}|\mathbf{m}, \mathbf{b}_h, \sigma_h) \quad (4)$$

The objective function can be defined as the expectation of the log-likelihood

$$E_{\mathbf{x}}[L_o] = -\frac{D}{2} \ln 2\pi - (D - H) \ln \sigma_0 - \sum_{h=1}^H \ln \sigma_h - \frac{1}{2} E_{\mathbf{x}} \left[\sum_{h=1}^H \left((\mathbf{x} - \mathbf{m})^T \mathbf{b}_h \right)^2 / \sigma_h^2 + \tilde{\mathbf{x}}^T \tilde{\mathbf{x}} / \sigma_0^2 \right] \tag{5}$$

2.2 Stochastic Optimization

An online-learning algorithm can be derived through stochastic gradient ascent on a sample function of (5), i.e. on $L_o(\mathbf{x})$. We have the partial derivatives

$$\frac{\partial L_o}{\partial \mathbf{m}} = \frac{1}{\sigma_0^2} \tilde{\mathbf{x}} + \sum_{h=1}^H \frac{(\mathbf{x} - \mathbf{m})^T \mathbf{b}_h}{\sigma_h^2} \mathbf{b}_h \tag{6}$$

$$\frac{\partial L_o}{\partial \mathbf{b}_h} = (\mathbf{x} - \mathbf{m})^T \mathbf{b}_h \left(\frac{1}{\sigma_0^2} \tilde{\mathbf{x}} - \frac{1}{\sigma_h^2} (\mathbf{x} - \mathbf{m}) \right) \tag{7}$$

$$\frac{\partial L_o}{\partial \sigma_h} = \frac{1}{\sigma_h} \left(\left(\frac{(\mathbf{x} - \mathbf{m})^T \mathbf{b}_h}{\sigma_h} \right)^2 - 1 \right) \tag{8}$$

$$\frac{\partial L_o}{\partial \sigma_0} = \frac{D - H}{\sigma_0} \left(\frac{\tilde{\mathbf{x}}^T \tilde{\mathbf{x}}}{(D - H) \sigma_0^2} - 1 \right) \tag{9}$$

In general, to obtain a learning algorithm, one then updates each parameter along the gradient for one step at a time. The stride at the n -th step is controlled by a positive learning rate $\lambda(n)$, which should satisfy $\sum_{n=0}^{\infty} \lambda(n) = \infty$ and $\sum_{n=0}^{\infty} \lambda^2(n) < \infty$ for convergence of the algorithm [12].

However, there are several issues to be considered carefully here. First, (8) and (9) can result in negative values of large magnitude, especially when σ_h or σ_0 is small, which would move σ_h or σ_0 to an invalid negative zone and lead to failure of the learning procedure. Second, direct maximization of the Gaussian likelihood function does not guarantee that $\mathbf{b}_h, h = 1, \dots, H$ be the *principal* eigenvectors, or \mathcal{L} be the *principal* subspace of the underlying distribution. In the following section, we will analyze these issues and obtain a robust solution.

2.3 The Local Linear Manifold Learning Rules

Let us investigate σ_h first. Ideally, $\sigma_h^2 = E_{\mathbf{x}}[(\mathbf{x} - \mathbf{m})^T \mathbf{b}_h]^2$. In practice, it can be estimated from n samples,

$$\sigma_h^2(n) = \frac{1}{n} \sum_{t=1}^n \left((\mathbf{x}(t) - \mathbf{m}(t))^T \mathbf{b}_h(t) \right)^2 \tag{10}$$

where $\mathbf{m}(t)$ and $\mathbf{b}_h(t)$ are the mean vector and the h -th basis vector at the t -th learning step. This equation is motivated by statistical efficiency [16]. An efficient estimator tends to converge most quickly and has the smallest error variance,

as demonstrated in [16]. For a Gaussian distribution with a known variance, the sample mean is an efficient estimator of the population mean. As pointed out in [16], the actual error variance is not very sensitive to the distribution. In (10) we may consider $\sigma_h^2(n)$ to be the sample mean of $w(t) = ((\mathbf{x}(t) - \mathbf{m}(t))^T \mathbf{b}_h(t))^2$. From (10) we can derive the following incremental relationship

$$\sigma_h^2(n) = \frac{n-1}{n} \sigma_h^2(n-1) + \frac{1}{n} \left((\mathbf{x}(n) - \mathbf{m}(n))^T \mathbf{b}_h(n) \right)^2 \tag{11}$$

Rearranging (11), we obtain

$$\sigma_h(n) - \sigma_h(n-1) = \frac{1}{n} \Delta \sigma_h(n) \tag{12}$$

where

$$\begin{aligned} \Delta \sigma_h(n) &= \frac{1}{\sigma_h(n) + \sigma_h(n-1)} \left(\left((\mathbf{x}(n) - \mathbf{m}(n))^T \mathbf{b}_h(n) \right)^2 - \sigma_h^2(n-1) \right) \\ &\approx \frac{\sigma_h(n-1)}{2} \left(\left(\frac{(\mathbf{x}(n) - \mathbf{m}(n))^T \mathbf{b}_h(n)}{\sigma_h(n-1)} \right)^2 - 1 \right) \end{aligned} \tag{13}$$

In (12), $\frac{1}{n}$ can be considered to play a role of the learning rate as in the usual stochastic gradient approaches. The expression (13) can be regarded as a discrete version of (8). Since it is expected that $\sigma_h^2 = E_{\mathbf{x}} [((\mathbf{x} - \mathbf{m})^T \mathbf{b}_h)^2]$, in the long run, the average updating based on both expressions tend to be zero. Their difference is basically a matter of step length. In (8) the step length is inversely proportional to σ_h , while in (13), it is proportional to σ_h , which is more reasonable and has a “multi-resolution” behavior. Furthermore, (13) avoids the problem of driving σ_h to negative since (13) is safely bounded below by $-\frac{1}{2}\sigma_h$.

The variance σ_0^2 can be considered as the average of variances in the rest $D-H$ directions other than the H principal directions, i.e. $\sigma_0^2 = E_{\mathbf{x}} [\tilde{\mathbf{x}}^T \tilde{\mathbf{x}}] / (D-H)$. Similar to σ_h^2 , we replace the expectation with a sample mean estimator

$$\sigma_0^2 = \frac{1}{n} \cdot \frac{1}{D-H} \sum_{t=1}^n \tilde{\mathbf{x}}^T(t) \tilde{\mathbf{x}}(t) \tag{14}$$

Following arguments similar to those of σ_h^2 , we can obtain the following discrete updating formula

$$\sigma_0(n) - \sigma_0(n-1) = \frac{1}{n} \Delta \sigma_0(n) \tag{15}$$

where

$$\Delta \sigma_0(n) \approx \frac{\sigma_0(n-1)}{2} \left(\frac{\tilde{\mathbf{x}}^T(n) \tilde{\mathbf{x}}(n)}{(D-H) \sigma_0^2(n-1)} - 1 \right) \tag{16}$$

Again, the right side of (16) has a form similar to that of (9). In the long run, both expressions tend to zero in average. The former has a more reasonable step length than the latter in the sense that the former is proportional to σ_0 . Also, (16) has a safe lower bound $-\frac{1}{2}\sigma_0$, which guarantees that the updating will not drive σ_0 to a negative zone.

To enforce a solution where \mathbf{b}_h span a principal subspace, the idea is to incorporate into (5) a regularization term proportional to $E[\tilde{\mathbf{x}}^T \tilde{\mathbf{x}}]$, since global minimization of this extra term leads to a principal subspace solution, which also coincides with the optimal solution of (5). We take partial derivatives of the sample function $\tilde{\mathbf{x}}^T \tilde{\mathbf{x}}$ with respect to \mathbf{m} and \mathbf{b}_h ,

$$\frac{\partial \tilde{\mathbf{x}}^T \tilde{\mathbf{x}}}{\partial \mathbf{m}} = -2\tilde{\mathbf{x}} \tag{17}$$

$$\frac{\partial \tilde{\mathbf{x}}^T \tilde{\mathbf{x}}}{\partial \mathbf{b}_h} = -2(\mathbf{x} - \mathbf{m})^T \mathbf{b}_h \cdot \tilde{\mathbf{x}} \tag{18}$$

Then we only need to combine (17), (18) with (6), (7) appropriately. In our practice we observed that (6) and (7) do not lead to a robust learning procedure. The reason is most likely related to the variances appearing in the denominators. Multiplying the derivatives with σ_0^2 , we obtain a form commonly used in the literature [1, 18]. Combining with $-\frac{1}{2}$ of the derivatives in (17) and (18), we obtain the following updating formulae

$$\Delta \mathbf{m}(n) = 2\tilde{\mathbf{x}}(n) + \sum_{h=1}^H \frac{\sigma_0^2(n-1)}{\sigma_h^2(n-1)} (\mathbf{x}(n) - \mathbf{m}(n-1))^T \mathbf{b}_{h(n-1)} \mathbf{b}_{h(n-1)} \tag{19}$$

$$\Delta \mathbf{b}_h(n) = (\mathbf{x}(n) - \mathbf{m}(n-1))^T \mathbf{b}_{h(n-1)} \left(2\tilde{\mathbf{x}}(n) - \frac{\sigma_0^2(n-1)}{\sigma_h^2(n-1)} (\mathbf{x}(n) - \mathbf{m}(n-1)) \right) \tag{20}$$

The final updating rules are obtained by introducing increments proportional to the gradients at each n -th step, more details will be given in the next section.

3 Topographically Ordered Invariant Feature Set Generation under a Self-organizing Framework

The procedure for learning the local linear manifold in the previous section defines the behavior of a single neuron. Such a linear manifold can be considered as a linearly invariant representation of the pattern class. A number of such neurons are then combined together under a self-organizing framework to account for nonlinear variation. The whole network then represents a set of topologically ordered invariant feature filters. This idea has been inspired by Kohonen’s pioneering work on the ASSOM [6]. Different from previous methods, in our model, we consider the local objective function from a likelihood point of view. Then at the network level, for the input vector $\mathbf{x}(n)$ at the n -th time instant, the winning neuron c is determined via competition

$$c = \arg \max_{q \in Q} L_q(\mathbf{x}(n)) \tag{21}$$

where $L_q(\mathbf{x}(n))$ is the sample likelihood function at neuron q , and Q is the set of neurons in the network.

The winner c and its neighbors $q \in Q$ then update their local linear manifolds following the stochastic optimization procedure developed in Section 2 to reflect new information received from the input $\mathbf{x}(n)$. The updating “force” of the neighbors of c are attenuated by a neighborhood function $\nu_{c,q}(n)$, which is a decreasing function of the distance between q and c . $\nu_{c,q}(n) = 1$ for $q = c$ and $0 \leq \nu_{c,q}(n) < 1$ otherwise. This defines a cooperative learning process which leads to topological ordering and well structuring of the neurons in the data space. For $q \neq c$, $\nu_{c,q}(n)$ should also be a decreasing function of the time variable n and $\nu_{c,q}(n) \rightarrow 0$ when $n \rightarrow \infty$, so that at the final stage of learning, each neuron only responds to inputs falling into its local “receptive field”. Each neuron $q \in Q$ updates its local linear manifolds according to the following formulae

$$\mathbf{m}_q(n) = \mathbf{m}_q(n-1) + \lambda(n)\nu_{c,q}(n)\Delta\mathbf{m}_q(n) \quad (22)$$

$$\mathbf{b}_{h,q}(n) = \mathbf{b}_{h,q}(n-1) + \frac{\lambda(n)\nu_{c,q}(n)}{\|\hat{\mathbf{x}}_q(n)\|\|\mathbf{x}(n)\|}\Delta\mathbf{b}_{h,q}(n) \quad (23)$$

$$\sigma_{h,q}(n) = \sigma_{h,q}(n-1) + \lambda(n)\nu_{c,q}(n)\Delta\sigma_{h,q}(n) \quad (24)$$

$$\sigma_{0,q}(n) = \sigma_{0,q}(n-1) + \lambda(n)\nu_{c,q}(n)\Delta\sigma_{0,q}(n) \quad (25)$$

where $h = 1, 2, \dots, H$, $\hat{\mathbf{x}}_q(n)$ is the projection of $\mathbf{x}(n)$ on the local linear manifold of the neuron q , $\Delta\mathbf{m}_q(n)$, $\Delta\mathbf{b}_{h,q}(n)$, $\Delta\sigma_{h,q}(n)$, and $\Delta\sigma_{0,q}(n)$ are defined for the neuron q according to (19), (20), (13), and (16), respectively. The learning rate parameter $\lambda(n)$ is often a $\frac{1}{n}$ -type function in stochastic optimization. In (23), $\|\hat{\mathbf{x}}_q(n)\|\|\mathbf{x}(n)\|$ in the denominator is used to normalize the updating step and improve the learning behavior [6]. At the beginning of learning, the angle between $\mathbf{x}(n)$ and the local linear manifold is large, and this term defines a large updating step. At the final stage of learning, $\mathbf{x}(n)$ tends to coincide with the local linear manifold, and this term defines a small updating step. The overall algorithm can be summarized as follows:

1. Initialize the list of parameters ($\mathbf{m}_q, \mathbf{b}_{h,q}, \sigma_{h,q}, \sigma_{0,q}$), e.g. randomly. The initial value $\sigma_{h,q}(0)$ can be set larger than $\sigma_{0,q}(0)$. $\mathbf{b}_{h,q}$, $h = 1, 2, \dots, H$ should be orthonormal for each neuron q in the network;
2. For the current input vector $\mathbf{x}(n)$, determine the winner c according to (21);
3. Update the winner c and those neurons q in its neighborhood according to (22)–(25). Orthonormalize the basis vectors $\mathbf{b}_{h,q}$ afterwards;
4. Repeat steps 2 and 3 until certain predetermined condition is satisfied, e.g. when the number of iterations reaches a predetermined maximum value.

This network is a self-organizing map which learns a set of topologically ordered local linear manifolds from input examples in an online fashion. The mean vector of each local linear manifold can be regarded as a prototype of the pattern class. The local linear manifold captures the most important directions of variation in the local partition. In this sense, each local linear manifold can be regarded as a feature filter invariant to some local linear transformations. The whole set of neurons then establish a globally nonlinear manifold, which is a global representation of the pattern class that is invariant to nonlinear transformations.

Furthermore, the whole set of neurons can be mapped to a low-dimensional topological display for convenient visualization and analysis, which is beyond the conventional manifold learning approaches, such as the ISOMAP or the LLE.

4 Experiments

In the following, we will demonstrate the capacity of the proposed network in generating invariant feature sets from visual patterns subject to substantial transformations through online learning. We will use handwritten digit image modeling and recognition as a specific example. It will be shown that recognition based on these representations are quite accurate.

4.1 Data and Related Work

A major difficulty in handwritten digit image modeling is the wide variety of writing styles affected by people and many other factors. The data set used in this paper is the MNIST database, which has been made publicly available by LeCun *et al.* [8] for evaluation of learning techniques and pattern recognition approaches on real-world data. This database is also used by Zheng *et al.* [18] in their experiments. Images in the MNIST database have been size normalized and centered in a 28×28 pixel field. Pixels of the resulting images are represented by gray levels due to the interpolation techniques used by the normalization procedure. Foreground pixels take high gray levels while background pixels take low gray levels. There are 60,000 training digit images and 10,000 test digit images. Figure 1 shows some representative digit images in this database. It is obvious that a wide variety of writing styles have been covered. Some of these digits are subject to substantial distortions, which poses considerable difficulty in accurate recognition. It is hard for hand-crafted feature extractors to deal with such extensive variation.

4.2 Invariant Feature Set Generation

The strategy that we have taken is to train a network for each digit. The images of each digit can be thought to be distributed along a nonlinear manifold with a dimensionality relatively lower than the dimensionality ($28 \times 28 = 784$ -D) of the original data space. Therefore, we build a self-organizing network for the underlying nonlinear manifold of each digit. Altogether, there are 10 of such



Fig. 1. Some examples from the MNIST database

networks, denoted by Q_k , $k = 0, 1, \dots, 9$ for the ten digits. Each network is composed of $r \times r$ neurons organized according to a rectangular topology. The H -dimensional local linear manifold \mathcal{L}_q at each neuron $q \in Q_k$ has a mean vector \mathbf{m}_q and H basis vectors $\mathbf{b}_{h,q}$, $h = 1, 2, \dots, H$. For each network, the number of learning steps is fixed to $N = 30,000$. The neighborhood function commonly takes the Gaussian form

$$\nu_{c,q}(n) = \exp\left(-\frac{\|\mathbf{u}_c - \mathbf{u}_q\|^2}{2\sigma_v^2(n)}\right) \quad (26)$$

where \mathbf{u}_c and \mathbf{u}_q are, respectively, the coordinates of the winning neuron c and an arbitrary neuron q in the network lattice. $\sigma_v(n) = \sigma_v(0)N/(N + 99n)$ is a variable that defines the scale of neighborhood, which shrinks with n to enforce ordering of neurons at the beginning of learning and more local learning at the final learning stage. $\sigma_v(0)$ should be appropriately set so that the whole network is covered by the full width at half maximum (FWHM) of the Gaussian neighborhood function at the beginning of learning. The learning-rate parameter

$$\lambda(n) = \lambda(0)\frac{N}{N + 99n} \quad (27)$$

for all the networks. The initial learning rate $\lambda(0) = 1$. Other settings did not show better results in our experiments.

Before a digit image is input into the networks, the mean value of its pixels is subtracted from each pixel of the image. The resulting image is then normalized to form a pattern vector \mathbf{x} to be input to the networks. As an example, the networks trained at $r = 4$ and $H = 3$ are shown in Fig. 2. Each network Q_k is composed of $4 \times 4 = 16$ neurons with each representing a 3-D local linear manifold defined by a mean vector \mathbf{m} and three orthonormal basis vectors \mathbf{b}_h , $h = 1, 2, 3$, whose components have been scaled to $[0, 255]$ for visualization. Therefore, the original $28 \times 28 = 784$ -D distribution of images of each digit is represented by $4 \times 4 = 16$ 3-D local linear manifolds, which globally construct a low-dimensional nonlinear manifold that is itself an invariant representation of the corresponding digit. Such a compact representation of the digit image distribution provides robustness to noise or over-fitting and generalization to future data. The local linear manifold learned by each neuron can be regarded as an invariant feature filter that captures linear transformations of up to three dimensions. Linear manifolds of the same digit are further organized according to a rectangular topology on a 2-D display for convenient visualization. Note that different “styles” of the handwritten digit images have been automatically generated. In our experiments, the generated visual patterns have appeared in less than 10 learning steps and become quite clear in about 100 learning steps.

4.3 Recognition Results

To evaluate the performance of the proposed approach in visual perception, we performed further experiments and compared the results to some previously

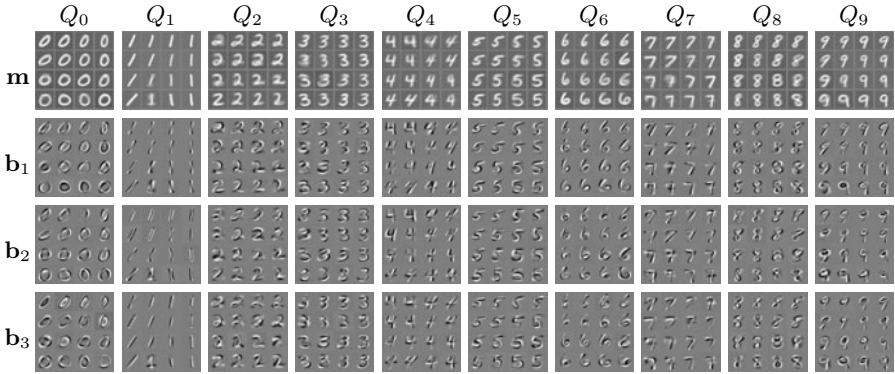


Fig. 2. Linear manifold self-organizing maps trained for the MNIST database

published similar models, including Kohonen *et al.*'s ASSOM [6] and Zheng *et al.*'s locally linear online mapping model (LLOM) [18]. Since each digit has been represented by a nonlinear manifold, or more specifically, a set of local linear manifolds in our case, we need to develop a measure of matching degree between input patterns and the nonlinear manifolds. If we consider each network as memory of the corresponding digit, the idea is to reconstruct a memorized pattern with information from the input pattern, and determine the difference between the reconstructed pattern and the input pattern. More specifically, for each input pattern vector \mathbf{x} of an unknown class from the test set, each of the 10 networks $Q_k, k \in \{0, 1, \dots, 9\}$ tries to reconstruct a memorized *closest* pattern $\hat{\mathbf{x}}_{Q_k}$ of its own. The network with the minimum reconstruction error determines the label of \mathbf{x} . The corresponding classification function can be defined as

$$l(\mathbf{x}) = \arg \min_{k \in \{0, 1, \dots, 9\}} \|\mathbf{x} - \hat{\mathbf{x}}_{Q_k}\| \tag{28}$$

where $\|\cdot\|$ corresponds to the usual Euclidean distance function.

Now it comes to the question of how to build the reconstruction $\hat{\mathbf{x}}_{Q_k}$. Recalling that the nonlinear manifold of the network Q_k is represented by a set of $r \times r$ local linear manifolds, we can combine the local linear reconstructions $\mathbf{m}_q + \hat{\mathbf{x}}_q$ ($q \in Q_k$) in a weighted way,

$$\hat{\mathbf{x}}_{Q_k} = \frac{\sum_{q \in Q_k} a_q (\mathbf{m}_q + \hat{\mathbf{x}}_q)}{\sum_{q \in Q_k} a_q} \tag{29}$$

where a_q is a weight parameter that should be relatively large for “good” local reconstructions. Such a strategy has also been adopted in [17] and [18]. There is no evidence that the functional form of a_q would be crucial to the performance [2]. A Gaussian function, which extends infinitely in the domain, has been chosen in the experiments, $a_q = \exp(-\|\mathbf{x} - \mathbf{m}_q - \hat{\mathbf{x}}_q\|^2 / (2\sigma_*^2))$, where σ_* is a parameter which controls the response field of the neurons. We have set $\sigma_* = 0.1$ in our experiments. It has been observed that this parameter can be

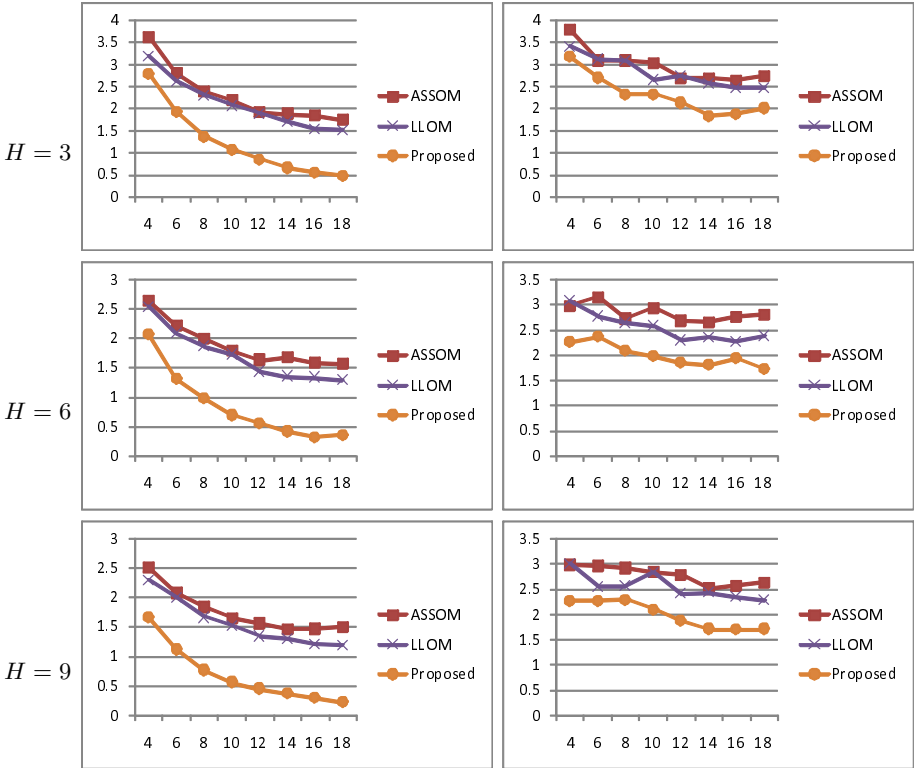


Fig. 3. Recognition results of different methods on the training set (left) and on the test set (right). In each subfigure, the vertical axis denotes the error rate (in percentage), the horizontal axis denotes the dimension r of each network.

chosen from a reasonably wide range of values without introducing significant difference to the results [17, 18].

The recognition results are plotted in Fig. 3, which shows that the proposed method has consistently lower error rates than the ASSOM and the LLOM under various configurations. In general, the performance of these methods improves with the local dimensionality H and the network size $r \times r$. On the training set, the proposed method reached a minimum error rate of 0.23% , which is lower than 1.47% of the ASSOM and 1.19% of the LLOM. On the test set, the proposed method reached a minimum error rate of 1.7% , which is also lower than 2.51% of the ASSOM and 2.27% of the LLOM. As a baseline comparison, the K -nearest neighbor classifier with a Euclidean distance measure shows an error rate of 5% on the test set [8]. However, memory access of the K -nearest neighbor classifier is much less efficient than the proposed method for large data sets. The proposed algorithm is quite stable in repeated running. For example, for 10 runs of 6×6 -sized networks with manifold dimension $H = 6$, the mean error rate (\pm standard deviation) is $1.26(\pm 0.03)\%$ on the training set and $2.10(\pm 0.14)\%$ on the test

set, respectively. For a comparison, under the same settings, the ASSOM and the LLOM have the error rates of $2.23(\pm 0.04)\%$ and $2.12(\pm 0.05)\%$ on the training set, $2.89(\pm 0.09)\%$ and $2.61(\pm 0.06)\%$ on the test set, respectively.

Different from usual manifold learning methods, the proposed algorithm provides abstraction and generalization of data. So it can tolerate reduction of the training set to some extent. For example, when trained on $\frac{1}{10}$ of the original training set, the error rate on the test set is 3.24% for 6×6 -sized networks with manifold dimension $H = 6$, which is still significantly better than that (5%) of the K -nearest neighbor classifier trained on the full-size training set.

5 Conclusions and Perspectives

This paper proposes a neural model which is able to learn a set of topologically ordered linear manifolds under a self-organizing framework. Each local linear manifold is an invariant feature filter that captures certain linear transformations. The whole network represents a globally nonlinear manifold embedded in the data space, which is an invariant representation of the target pattern subject to nonlinear transformations. Compared to other similar models, such as the ASSOM or those models in [1] and [18], the proposed model has some extra parameters, i.e. standard deviations, which are only one-dimensional and do not increase the complexity much. On the other hand, the proposed model is more adaptive to the data under study, as verified by the performance in handwritten digit image modeling and recognition. Compared to conventional manifold learning approaches, the proposed model provides abstraction and generalization of data, which is important to tasks related to visual perception.

It has been observed in our experiments of handwritten digit recognition that although the proposed model shows very remarkable performance on the training set (error rate of as small as 0.23%), it can not approach such a level on the test set. Some kind of overfitting to the training set seems to have occurred. An interesting direction of further study is to develop new methods to explore this “gap” and balance learning on the training set and generalization on the test set appropriately.

Acknowledgement. This work was supported by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry, the Specialized Research Fund for the Doctoral Program of Higher Education (200805581005), the Fundamental Research Funds for the Central Universities (09lgpy52), and the Innovative Research Fund for Outstanding Young Scholars of Guangdong Higher Education.

References

1. Adibi, P., Safabakhsh, R.: Linear manifold topographic map formation based on an energy function with on-line adaptation rules. *Neurocomputing* 72, 1817–1825 (2009)
2. Atkeson, C.G., Moore, A.W., Schaal, S.: Locally weighted learning. *Artificial Intelligence Review* 11, 11–73 (1997)

3. Cai, D., He, X., Hu, Y., Han, J., Huang, T.: Learning a spatially smooth subspace for face recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–7 (2007)
4. Kohonen, T.: The adaptive-subspace SOM (ASSOM) and its use for the implementation of invariant feature detection. In: Proceedings of International Conference on Artificial Neural Networks, pp. 3–10 (1995)
5. Kohonen, T.: Self-Organizing Maps, 3rd edn. Springer, Heidelberg (2001)
6. Kohonen, T., Kaski, S., Lappalainen, H.: Self-organized formation of various invariant-feature filters in the adaptive-subspace SOM. *Neural Computation* 9, 1321–1344 (1997)
7. Laaksonen, J., Oja, E.: Subspace dimension selection and averaged learning subspace method in handwritten digit classification. In: Vorbrüggen, J.C., von Seelen, W., Sendhoff, B. (eds.) ICANN 1996. LNCS, vol. 1112, pp. 227–232. Springer, Heidelberg (1996)
8. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 2278–2324 (1998)
9. Liu, Z.Q.: Adaptive subspace self-organizing map and its application in face recognition. *International Journal of Image and Graphics* 2, 519–540 (2002)
10. López-Rubio, E., Muñoz-Pérez, J., Gómez-Ruiz, J.A.: A principal components analysis self-organizing map. *Neural Networks* 17, 261–270 (2004)
11. Oja, E.: Subspace Methods of Pattern Recognition. Research Studies Press, Letchworth (1983)
12. Robbins, H., Monro, S.: A stochastic approximation method. *The Annals of Mathematical Statistics* 22, 400–407 (1951)
13. Roweis, S., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. *Science* 290, 2323–2326 (2000)
14. Seung, H.S., Lee, D.D.: The manifold ways of perception. *Science* 290, 2268–2269 (2000)
15. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. *Science* 290, 2319–2323 (2000)
16. Weng, J., Zhang, Y., Hwang, W.S.: Candid covariance-free incremental principal component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25, 1034–1040 (2003)
17. Zhang, B., Fu, M., Yan, H., Jabri, M.: Handwritten digit recognition by adaptive-subspace self-organizing map (ASSOM). *IEEE Transactions on Neural Networks* 10, 939–945 (1999)
18. Zheng, H., Shen, W., Dai, Q., Hu, S., Lu, Z.M.: Learning nonlinear manifolds based on mixtures of localized linear manifolds under a self-organizing framework. *Neurocomputing* 72, 3318–3330 (2009)

A Multi-level Supporting Scheme for Face Recognition under Partial Occlusions and Disguise

Jacky S-C. Yuk, Kwan-Yee K. Wong, and Ronald H-Y. Chung

Dept. of Computer Science
The University of Hong Kong, Hong Kong

Abstract. Face recognition has always been a challenging task in real-life surveillance videos, with partial occlusion being one of the key factors affecting the robustness of face recognition systems. Previous researches had approached the problem of face recognition with partial occlusions by dividing a face image into local patches, and training an independent classifier for each local patch. The final recognition result was then decided by integrating the results of all local patch classifiers. Such a local approach, however, ignored all the crucial distinguishing information presented in the global holistic faces. Instead of using only local patch classifiers, this paper presents a novel multi-level supporting scheme which incorporates patch classifiers at multiple levels, including both the global holistic face and local face patches at different levels. This supporting scheme employs a novel criteria-based class candidates selection process. This selection process preserves more class candidates for consideration as the final recognition results when there are conflicts between patch classifiers, while enables a fast decision making when most of the classifiers conclude to the same set of class candidates. All the patch classifiers will contribute their supports to each selected class candidate. The support of each classifier is defined as a simple distance-based likelihood ratio, which effectively enhances the effect of a “more-confident” classifier. The proposed supporting scheme is evaluated using the AR face database which contains faces with different facial expressions and face occlusions in real scenarios. Experimental results show that the proposed supporting scheme gives a high recognition rate, and outperforms other existing methods.

1 Introduction

Over the last decade, many mature algorithms have been developed for face recognition [1, 2, 3, 4, 5]. These algorithms often demonstrate promising results with high recognition rates on face image captured under ideal conditions such as frontal faces in passport photos. On the other hand, face recognition has always been a challenging problem in real-life surveillance videos where faces are always non-frontal, occluded, and in low resolutions. In particular, recognizing partially occluded or disguised faces is one of the key issues in enhancing the robustness of face recognition in real-life videos.

Currently, there are not many effective and efficient methods to handle face recognition with occlusions. Some common approaches to tackle the problem include face occlusion detection [6,7,8,9] and face division into local patches [9,10,11,12,13].

In [6], the occluded parts of a face were first detected according to the residual values, and a new face classifier was trained using the training samples with all the occluded parts being masked-out. Although this approach can effectively ignore the effect of the occluded parts, the recognition is extremely time-consuming since a new classifier has to be trained in run-time for every recognition. Fidler et al. [8] proposed a subspace recovering method for recovery the faces from occlusions. Their method reconstructed occluded parts of a face from the trained subspace before performing the recognition. The recognition correctness, however, is lowered due to the recovery errors, especially when the individual is not included in the training set. Jia and Martinez [14] suggested to use faces with occlusions as training samples to train a SVM classifier. This approach, however, is risky when the occlusion scenarios are not included in the training samples. Oh et al. [9] proposed a selective-LNMF classifier. Their method first divides and locates the occluded face patches, and re-projects the training samples to the selective-LNMF space, in which the LNMF bases belonging to the occluded face patches are excluded. The recognition stage of this method, however, can be very time-consuming when the face database grows large. Moreover, this method requires occlusion detection which was trained by partially occluded face samples, therefore, the method cannot solve the unseen occlusion case.

Instead of using occlusion detection and face recovery, Martinez [11] suggested to divide a face into 6 local patches, and weight each local patch according to a new training face set. This method then votes for the final recognition results according to the weightings of the local patches. Such a local face patches approach enhances the face recognition rate since it reduces the effect of the occluded parts in the recognition. However, the distinguishing information in the holistic face is also crucial in face recognition. If only local face patches are considered, the distinguishing information of the holistic face may be ignored. Kim et al. [13] suggested to combine local features and global holistic face information in the recognition. In their method, local-feature patches, including eyes, nose, mouth, are first located by local feature detectors. The final recognition is then decided by combining the local and global holistic face recognition results. They showed that their combination method outperforms both the global holistic approach as well as the local-patch approach. However, Kim et al. did not elaborate their method on occluded faces where the local face features might be occluded, and might not be easy to locate.

This paper proposes a novel multi-level supporting scheme which integrates the recognitions of global holistic face and multi-level local face patches. The main contributions of this paper include: 1) a novel multi-level supporting scheme which incorporates the decisions multi-level patch classifiers, 2) a simple and effective distance-based likelihood ratio to enhance the weightings of “more-confident” patch classifiers, and 3) a criteria-based class candidates selection

process which preserves more class candidates for consideration as the final recognition result when there are conflicts between patch classifiers. In summary, our method first divides a face image into local patches at different levels (figure 1). For each patch, including the global face image, a fisherface subspace classifier [15] is trained. In the testing stage, a testing face image is also divided into local patches as in the training stage. A multi-level supporting scheme is then applied to integrate the recognition results of the local patches. The scheme first selects potential class candidates according to the matching likelihood ratio between the testing and training faces. Each local patch classifier is then invited to give its support to these selected candidates. The final recognition result is decided according to the supports from all patch classifiers. The proposed scheme is efficient since it requires neither re-training nor re-projection of the training faces. The supporting scores contributed by the patch classifiers depend on a simple likelihood ratio which will be discussed in detail in Section 2. The proposed likelihood ratio measures how likely a testing patch belongs to the same class of a particular training face patch, and effectively decreases the effects of those patches with low confidence. Furthermore, the discriminant information on multi-level patches, including the global holistic face and local smaller patches, are all being considered and integrated. The proposed recognition is, therefore, more robust to partial face occlusions and facial expression changes. The proposed scheme is evaluated using the AR face database [16] which contains faces with different facial expressions and real occlusions. Experimental results shown in Section 3 shows the proposed scheme gives a high recognition rate, and generally outperforms existing state-of-the-art methods.

The paper is organized as follows. Section 2 describes in detail the proposed multi-level supporting face recognition scheme. Experimental results are then presented in Section 3, followed by the conclusions in Section 4.

2 Multi-level Supporting Scheme

Face images are first divided into patches at different levels with slight overlapping (about one-eighth of the width/height) as shown in figure 1. In the experiments presented in this paper, each face image is divided into 2x2, 4x1, 1x4, 4x2 and 2x4 patches. Together with the original holistic 1x1 face image, there are in total 29 image patches. For each image patch, an independent classifier is trained as described in following sections.

2.1 Fisherface Subspace Classifiers

This section describes the subspace classifier for a single image patch. The classifiers for all the other image patches, including the global holistic face patch, are trained in the same way. For each face image patch, an independent fisherface classifier [15] is trained. Suppose there are N training face sample. An image patch of the i -th training sample is represented as a 1-D vector \mathbf{x}_i in single grey channel. The vector \mathbf{x}_i is projected to an eigenface subspace using principle component analysis (PCA) [15]:

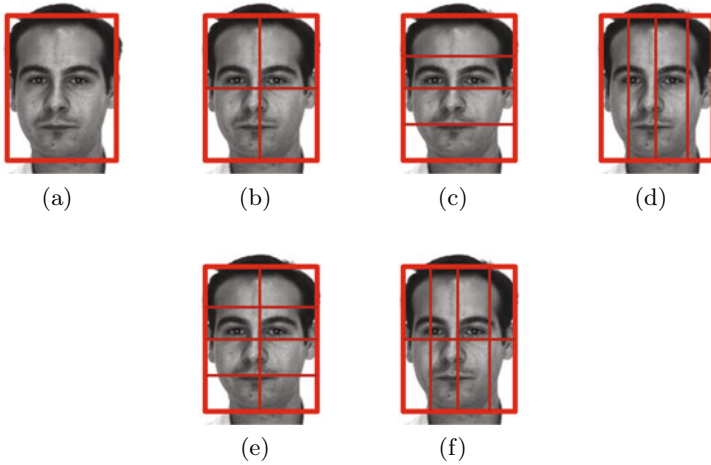


Fig. 1. Faces are divided into slightly overlapped patches at different levels: (a) manually cropped 1x1 holistic face, (b) 2x2 face patches, (c)(d) horizontal 4x1 and vertical 1x4 face patches, and (e)(f) horizontal 4x2 and vertical 2x4 face patches

$$\hat{x}_i = U_K^T(x_i - m) \tag{1}$$

where m is the mean vector of all training patch vectors x , $U_K = [u_1, \dots, u_K]$ is a matrix whose columns are the K eigenvectors with the largest eigenvalues of the scatter matrix S_T :

$$S_T = \sum_{i=1}^N (x_i - m)(x_i - m)^T \tag{2}$$

The set containing N faces in the fisherface subspace $\hat{Y} = \{\hat{y}_1, \dots, \hat{y}_N\}$ is then constructed by projecting the corresponding \hat{x}_i to the fisherface subspace using linear discriminant analysis (LDA):

$$\hat{y}_i = W^T(\hat{x}_i - \hat{m}) \tag{3}$$

where \hat{m} is the mean vector of all training patch vectors \hat{x} in PCA subspace. W contains the bases of the LDA subspace which is calculated by maximizing the between-class scatter matrix S_b and minimizing the within-class scatter matrix S_w . The optimal W_{opt} is defined as:

$$W_{opt} = arg \ max \left| \frac{W^T S_b W}{W^T S_w W} \right| \tag{4}$$

$$S_b = \sum_{i=1}^C n_i (\hat{m}_i - \hat{m})(\hat{m}_i - \hat{m})^T \tag{5}$$

$$S_w = \sum_{i=1}^C \sum_{\hat{x}_k \in \hat{X}_i} (\hat{x}_k - \hat{m}_i)(\hat{x}_k - \hat{m}_i)^T \tag{6}$$

where C is the total number of training classes. n_i is the number of samples of the i -th class. \hat{m}_i and \hat{m} are the mean of the i -th class and the mean of all PCA samples respectively, and $\hat{X}_i = \{\hat{x}_k\}$ contains all PCA samples in the i -th class. As suggested in [1], this paper directly calculates the optimal $W_{opt} = [w_1, \dots, w_{\hat{K}}]$ as the first \hat{K} eigenvectors of $S_w^{-1}S_b$ with the largest eigenvalues.

2.2 Matching Likelihood Ratio

During the training stage, the mean μ^{intra} and variance ν^{intra} of the intra-class distances are calculated as:

$$\mu^{intra} = \frac{1}{N^{intra}} \sum_{c_k=1}^C \sum_{\hat{y}_i \in \hat{Y}_{c_k}} \sum_{\hat{y}_j \in \hat{Y}_{c_k}}^{i < j} d_{i,j} \tag{7}$$

$$\nu^{intra} = \frac{1}{N^{intra}} \sum_{c_k=1}^C \sum_{\hat{y}_i \in \hat{Y}_{c_k}} \sum_{\hat{y}_j \in \hat{Y}_{c_k}}^{i < j} (d_{i,j} - \mu^{intra})^2 \tag{8}$$

where C is the total number of classes, $d_{i,j} = [(\hat{y}_i - \hat{y}_j)^T \Sigma^{-1}(\hat{y}_i - \hat{y}_j)]^{1/2}$ is the Mahalanobis distance between \hat{y}_i and \hat{y}_j , $\hat{Y}_c = \{\hat{y}_i : \hat{y}_i \in class\ c\}$ contains all the faces of class c in fisherface subspace, and N^{intra} is the total number of the intra-class combinations.

Similarly, the mean μ^{inter} and variance ν^{inter} of inter-class distances are defined as:

$$\mu^{inter} = \frac{1}{N^{inter}} \sum_{c_k=1}^C \sum_{\hat{y}_i \in \hat{Y}_{c_k}} \sum_{\hat{y}_j \in \hat{Y}_{c_t}}^{c_k < c_t \leq c_N} d_{i,j} \tag{9}$$

$$\nu^{inter} = \frac{1}{N^{inter}} \sum_{c_k=1}^C \sum_{\hat{y}_i \in \hat{Y}_{c_k}} \sum_{\hat{y}_j \in \hat{Y}_{c_t}}^{c_k < c_t \leq c_N} (d_{i,j} - \mu^{inter})^2 \tag{10}$$

where N^{inter} is the number of inter-class combinations.

With the means and variances of intra- and inter-class distances, the matching likelihood ratio $L_{i,j}$ is defined based on the distance $d_{i,j}$ between \hat{y}_i and \hat{y}_j :

$$L_{i,j} = \frac{p^{intra}(d_{i,j})}{p^{inter}(d_{i,j})} \tag{11}$$

where $p^{intra}(d)$ and $p^{inter}(d)$ are the probability density functions (pdf) of intra- and inter-class distances respectively. $p^{intra}(d)$ and $p^{inter}(d)$ are implemented as a slightly modified Gaussian functions:

$$p^{intra}(d) = \frac{1}{\sqrt{2\pi\nu^{intra}}} e^{-\frac{(t^{intra}-\mu^{intra})^2}{2\nu}} \tag{12}$$

$$p^{inter}(d) = \frac{1}{\sqrt{2\pi\nu^{inter}}} e^{-\frac{(t^{inter}-\mu^{inter})^2}{2\nu}} \tag{13}$$

where μ^{intra} and ν^{intra} are intra-class distance mean and variance specified in (7) and (8) respectively, and μ^{inter} and ν^{inter} are inter-class distance mean and variance specified in (9) and (10) respectively. $t^{intra} = \max(d, \mu^{intra})$ and $t^{inter} = \min(d, \mu^{inter})$ are the modified distance terms for p^{intra} and p^{inter} respectively. As illustrated in figure 2, these two terms ensure the likelihood ratio L obeys the similarity rule. The distance d is assumed to give equal intra-class probability $p^{intra}(d)$ when $d < \mu^{intra}$, and give equal inter-class probability $p^{inter}(d)$ when $d > \mu^{inter}$.

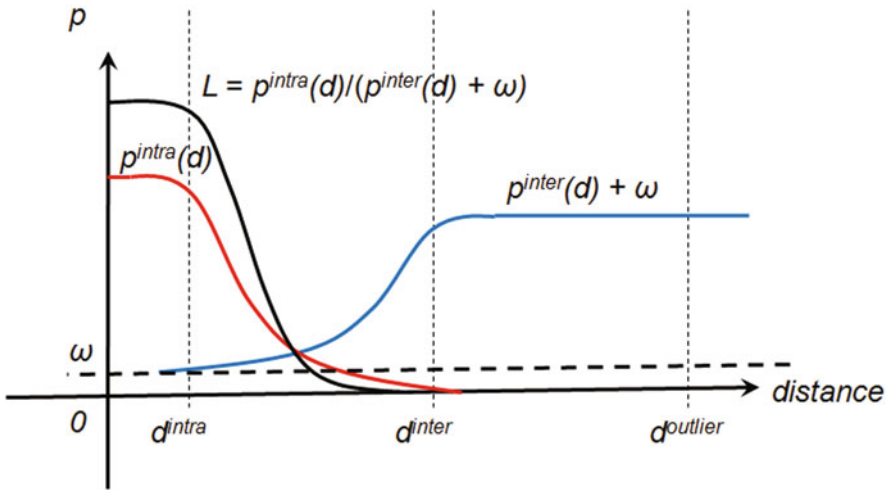


Fig. 2. An illustration of the likelihood functions. The likelihood L is large when the distance belongs to intra-class distance (d^{intra}), and L decreases dramatically when the distance approaches the inter-class distance (d^{inter}) or the outlier distance ($d^{outlier}$).

Given the distance $d_{i,j}$, $p^{intra}(d_{i,j})$ measures how likely \hat{y}_i and \hat{y}_j belong to the same class, whereas $p^{inter}(d_{i,j})$ measures how likely these two faces belong to different classes. Therefore, the larger the likelihood ratio defined in (11), the more likely the faces belong to the same class. Furthermore, an ω term is also added in the denominator of (11) to fix the likelihood ratio L :

$$L_{i,j} = \frac{p^{intra}(d_{i,j})}{p^{inter}(d_{i,j}) + \omega} \tag{14}$$

This ω term is used to prevent the likelihood ratio $L_{i,j}$ of a particular patch classifier becoming too large when the corresponding distance $d_{i,j}$ is too small, and therefore, preventing such patch classifier dominating the final recognition result. As illustrated in figure 2, this formulation effectively enhances the likelihood ratio when a face patch is matching with an intra-class patch, and the ratio decreases dramatically when the face patch is matching with an inter-class patch or an outlier/occluded patch with reasonable assumption that $d^{intra} < d^{inter} < d^{outlier}$.

2.3 Class Candidates Selection

In the recognition stage, a testing face image is divided into patches in the same way as the training images shown in figure 1. Each patch then undergoes classification matchings with the corresponding patches of the training samples. For a patch classifier p , the matching likelihood ratio $L_{p,c,k}$ of the k -th training sample in class c is calculated as in (14). After that, a set of class candidates is selected based on a criteria-based majority voting. The class candidate set is constructed in two stages: 1) First, a set of class votes $\mathbf{V} = \{v_c\}$ is constructed by a criteria-voting, where v_c is the number of votes for class c . Each patch classifier p votes for c whenever there exists a training sample k belonging to c with a matching likelihood ratio $L_{p,c,k}$ larger than a pre-defined threshold τ . 2) The class candidate set $\hat{\mathbf{C}}$ is then constructed as:

$$\hat{\mathbf{C}} = \{c : v_c > \lambda\} \quad (15)$$

where λ is a loose-to-fine variable threshold. In the experiment, τ is set to 0.9, and λ is set to $M/2$ at first where M is the total number of patch classifiers. λ is then iteratively decreased by halving its value at each step until $\hat{\mathbf{C}} \neq \emptyset$. This variable λ preserves more class candidates when there is more conflicts between classifiers. On the other hand, a faster decision can be made when majority of the classifiers are supporting to certain classes.

2.4 Multi-level Supporting

For each potential class candidate selected, the supporting is initiated by asking the support $s_{p,c}$ for the corresponding class c from each patch classifier p . The support from the p -th patch classifier is simply defined as the maximum likelihood ratio of the samples belonging to class c :

$$s_{p,c} = \max_k L_{p,c,k} \text{ for all sample } k \in \text{class } c \quad (16)$$

The final support S_c for a class c is then defined as the weighted sum of $s_{p,c}$:

$$S_c = \sum \alpha_p s_{p,c} \quad (17)$$

where α_p is the corresponding weighting of the patch classifier p . In the experiment, the weightings α_p of all patch classifiers are set to equal-value, and so the supports from all classifiers are equally weighted.

3 Experimental Results

The proposed method is evaluated using the AR database [16] with real occlusion scenarios and different facial expressions. The database contains 134 individuals including 76 males and 58 females. For each individual, there are several face categories in which faces are in different facial expressions and occlusions (figures 3). In the experiments, the face categories normal (figure 3(a)(g)), smile (figure 3(b)(h)) and angry (figure 3(c)(i)) are used for training. The face categories scream (figure 3(d)(j)), sun-glasses (figure 3(e)(k)) and scarf (figure 3(f)(l)) are used for testing the proposed scheme with real occlusions and in different facial expressions. In addition, the normal face category is made synthetically occluded by random masks (figure 4). This set is used for evaluating the proposed scheme under synthetic occlusions. All the faces for training and testing are manually cropped, aligned by eyes, and resized to 48x64.

3.1 Synthetic Occlusions

The faces in the normal category were occluded by synthetic black masks at random positions as shown in figure 4. The dimensions of these black boxes were also randomly selected with approximate size of 16%, 25%, 36%, 49% and 64% of the whole face image respectively. The occluded face images were then used to evaluate the proposed method.

Table 1. Face recognition results with synthetic occlusion masks

	Recognition Rate (%)				
	16% Occl.	25% Occl.	36% Occl.	49% Occl.	64% Occl.
Prop. ML-Support	100.0	100.00	98.51	90.30	72.39
Local-Vote(4x2)	100.0	98.51	84.33	67.91	53.73
ML-Vote	100.0	97.01	76.12	56.72	38.06
Fisher [15]	88.06	56.72	26.87	14.93	7.46

Table 1 shows the recognition results of the proposed multi-level supporting scheme (Prop. ML-Support). The recognition results of fisherface (Fisher) [15], majority voting of local patches (Local-Vote) and majority voting of patches at all levels (ML-Vote) are also listed in the table. Note that the testing samples are selected from one of the face categories used for training with synthetic occlusions added. The Local-Vote approach takes the advantages under heavy occlusions since the non-occluded patches should match exactly with the corresponding training patches, and thus outperforms the ML-Vote approach whose results are affected by the patches in the higher levels under heavy occlusions. The proposed supporting scheme, on the other hand, is able to enhance the leverage of the non-occluded patch classifiers, and incorporate those “more-confident” patches at different levels. The results show that the proposed scheme outperforms the fisherface and other majority voting approaches, and is able to enhance the recognition rate up to nearly 90% and 72% under extremely heavy occlusions of about 49% and 64%, respectively.

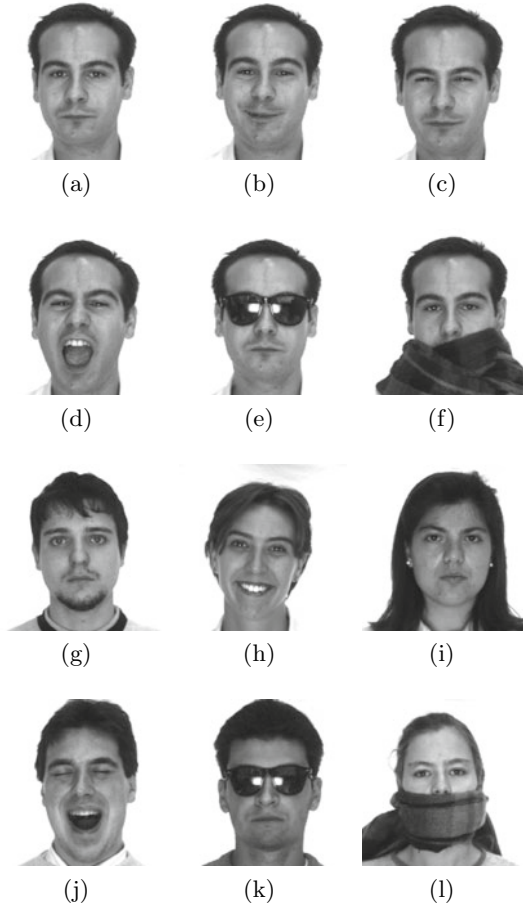


Fig. 3. Face samples in the AR database which contains faces with different facial expressions and occlusion scenarios. The first two rows show the faces of a particular individual in different face categories. The next two rows show other individuals' faces which belong to the corresponding categories as the first two rows. The face categories include: (a)(g) normal, (b)(h) smile, (c)(i) angry, (d)(j) scream, (e)(k) sun-glasses and (f)(l) scarf.

3.2 Facial Expression Changes and Real Occlusions

Table 2 lists the recognition rates of the proposed Multi-Level supporting scheme (Prop. ML-Support) with real occlusion scenarios (figure 3(e)(k) sun-glasses and (g)(l) scarf) and different facial expressions (figure 3(d)(j) scream). Similar to the synthetic occlusion experiments, the recognition results of fisherface (Fisher) [15], majority voting of local patches (Local-Vote) and majority voting of patches at all levels (ML-Vote) are also listed in the table. Furthermore, the recognition rates presented in [8] (Sub-Recovery), [14] (Occl-SVM) and [9] (sLNMF) are also included for comparisons. The results show that the proposed supporting scheme



Fig. 4. Cropped faces in category normal with synthetic occlusions of about: (a)(f) 16%, (b)(g) 25%, (c)(h) 36%, (d)(i)49% and (e)(j)64%

Table 2. Face recognition results with real occlusion

	Recognition Rate (%)		
	Scream	Sun-glasses	Scarf
Prop. ML-Support	92.54	92.54	93.28
Local-Vote(4x2)	88.81	82.09	92.54
ML-Vote	90.03	85.07	89.55
Fisher [15]	67.16	59.70	32.84
Sub-Recovery [8]	87.00	84.00	93.00
Occl-SVM [14]	–	57.0	57.0
sLNMF [9]	44	90	92

outperforms the traditional holistic and majority voting approaches under real occlusions and facial expression changes.

The performance of the proposed scheme is also generally better than the previous methods [8, 14, 9]. Unlike sLNMF [9], the proposed scheme not only tackles recognition under partial occlusions, but also tolerates facial expression changes. The recognition rate of the proposed scheme is much better than Jia and Martinez’s method (Occl-SVM) [14]. Note that Jia and Martinez used the occluded faces (sun-glasses and scarf categories) as training samples, and another set of sun-glasses and scarf face categories, which were taken separately, is used as testing samples. It is expected that the results of the proposed supporting scheme will be even better if such occluded face sets are also used for the training. The proposed scheme demonstrates slightly better results than Fidler et al.’s recovery approach [8] for the scarf samples, and outperforms their method for the scream and sun-glasses samples. Note that the proposed scheme does not require complicated iterative face recovery process, and therefore, is more efficient.

4 Conclusions

This paper introduces a novel multi-level supporting scheme for face recognition under partial occlusions and disguise. This scheme effectively incorporates the face discriminant information at multiple face levels with the proposed matching likelihood ratio for each face patch. This likelihood ratio is designed to enhance the effect of well-matched patches while making the effect of bad-matched patches negligible. This approach allows the best-matched patch classifiers to give more contributions since they are the “most-confident” classifiers. In addition, the candidate selection scheme also allows more individual candidates to be considered at the initial recognition stage when there exist conflicting classifiers, and thus enhancing the final supporting results. Experimental results show the proposed method provides a more robust and effective face recognition system, especially when the faces are under occlusions, and it can tolerate different facial expressions. The results also demonstrate the proposed method outperforms the previous methods under such scenarios.

References

1. Etemadm, K., Chellappa, R.: Discriminant analysis for recognition of human face images. *Journal of the Optical Society of America A* 14, 1724–1733 (1997)
2. Moghaddam, B., Jebara, T., Pentland, A.: Bayesian face recognition. *Pattern Recognition* 33, 1771–1782 (2000)
3. Guo, G., Li, S.Z., Chan, K.: Face recognition by support vector machines. In: *Proc. IEEE Int. Conf. on Automatic Face and Gesture Recognition*, Grenoble, France, pp. 196–201 (2000)
4. Li, Z., Tang, X.: Bayesian face recognition using support vector machine and face clustering. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Washington D.C., USA, pp. 374–380 (2004)
5. Wang, X., Tang, X.: Random sampling for subspace face recognition. *International Journal of Computer Vision (IJCV)* 70, 91–104 (2006)
6. Lanitis, A.: Person identification from heavily occluded face images. In: *Proc. 2004 ACM Symposium on Applied Computing (SAC 2004)*, pp. 5–9. ACM, New York (2004)
7. Lin, D., Tang, X.: Quality-driven face occlusion detection and recovery. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA, pp. 1–7 (2007)
8. Fidler, S., Skocaj, D., Leonardis, A.: Combining reconstructive and discriminative subspace methods for robust classification and regression by subsampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28, 337–350 (2006)
9. Oh, H.J., Lee, K.M., Lee, S.U., Yim, C.-H.: Occlusion invariant face recognition using selective LNMF basis images. In: Narayanan, P.J., Nayar, S.K., Shum, H.-Y. (eds.) *ACCV 2006. LNCS*, vol. 3851, pp. 120–129. Springer, Heidelberg (2006)
10. Heisele, B., Ho, P., Poggio, T.: Face recognition with support vector machines: Global versus component-based approach. In: *Proc. Eighth IEEE Int. Conf. on Computer Vision (ICCV 2001)*, Vancouver, Canada, pp. 688–694 (2001)
11. Martinez, A.M.: Recognizing imprecisely localized, partially occluded and expression variant faces from a single sample per class. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 748–763 (2002)

12. Ekenel, H.K., Stiefelhagen, R.: Local appearance based face recognition using discrete cosine transform. In: Proc. 13th European Signal Processing Conf. (EUSIPCO 2005), Antalya, Turkey (2005)
13. Kim, C., Oh, J., Choi, C.: Combined subspace method using global and local features for face recognition. In: Proc. IEEE Int. Joint Conf. Neural Networks, Montreal, Canada, vol. 4, pp. 2030–2035 (2005)
14. Jia, H., Martinez, A.M.: Support vector machines in face recognition with occlusions. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, pp. 136–141 (2009)
15. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. In: Buxton, B.F., Cipolla, R. (eds.) ECCV 1996. LNCS, vol. 1064, pp. 45–58. Springer, Heidelberg (1996)
16. Martinez, A.M., Benavente, R.: The AR face database. CVC Tech. Report 24 (1998)

Foreground and Shadow Segmentation Based on a Homography-Correspondence Pair

Haruyuki Iwama, Yasushi Makihara, and Yasushi Yagi

Osaka University, 8-1 Mihogaoka, Ibaraki, Osaka, 567-0047, Japan

Abstract. A static binocular camera system is widely used in many computer vision applications; and being able to segment foreground, shadow, and background is an important problem for them. In this paper, we propose a homography-correspondence pair-based segmentation framework. Existing segmentation approaches, based on homography constraints, often suffer from occlusion problems. In our approach, we treat a homography-correspondence pair symmetrically, to explicitly take the occlusion relationship into account, and we regard the segmentation problem as a multi-labeling problem for the homography-correspondence pair. We then formulate an energy function for this problem and get the pair-wise segmentation results by minimizing them via an α - β swap algorithm. Experimental results show that accurate segmentation is obtained in the presence of the occlusion region in each side image.

1 Introduction

In many computer vision applications, foreground segmentation is important preprocessing for subsequent processing such as object detection, localization, identification and tracking. For this purpose, background subtraction has been widely used for scenes with a static camera [1]. The methods, however, often extract not only the objects but also their shadows, which can be problematic. Consequently, many shadow segmentation, detection, or removal techniques have been proposed [2] [3] [4] [5] [6] [7] [8] [9], based mainly on the following two properties of shadow color: (a) The shadow region is darker than the original background region, (b) The color vector direction of the shadow region is similar to that of the original background region. The property (b) is not reliable in the case of a strong shadow because the color vector direction becomes unstable, and furthermore, it is not true under multiple-color illumination conditions, such as an outdoor scene with daylight. Eventually, any state of the art method using a color based approach fails if the foreground regions contain exactly the same color as the shadows, such as a black-haired human head. This is the essential limitation of color-based segmentation.

On the other hand, a static binocular camera system is widely used in many computer vision applications, such as surveillance systems, traffic monitoring systems, and total coverage camera systems in soccer stadiums. Taking advantage of the multi-view framework, several geometric approaches have been applied to the

foreground/shadow segmentation. One well known approach is foreground separation from shadow based on disparities [10] [11]. However, it often suffers from mis-correspondence problems and cannot be applied to scenes with no texture.

Alternatively, a homography constraint is also popular as a geometric constraint between multiple viewpoints [12] [13] [14] [15]. Approaches based on homography aim mainly to distinguish standing objects from ground plane objects including shadow. Their mechanism is described as follows. Suppose that one side image is a base image to be processed. If the color of a pixel in the base side image is similar to that of the homography-correspondence pixel in the other side image, the pixel in the base side image is regarded as part of the ground plane objects, and vice versa. Although the homography approach is quite effective despite the low computational cost, most of the existing methods, however, have a serious disadvantage. Let us consider the following case: A pixel belongs to the ground plane objects in the base side image and the homography-correspondence pixel is occluded by a foreground object in the other side image. In such a case, because the colors of the pair of pixels are different, the pixel in the base image is incorrectly regarded as a foreground object.

In the field of stereo correspondence problems, symmetric correspondence based approaches have been proposed to handle such occlusions appropriately [16] [17]. These approaches explicitly take the occlusion relationship into account by treating a stereo correspondence pair in a symmetric way.

Inspired by the symmetric approaches, we propose a symmetric segmentation framework based on a homography constraint with occlusion handling. Our goal is “*how to segment foreground, shadow, and background*”, and we regard this segmentation problem as a homography-correspondence pair labeling problem. Then, we solve this in an energy minimization framework together with a graph-cut algorithm [18]. Considering the homography-correspondence symmetrically, we cannot only segment the occluded region correctly, but also acquire additional information about the occluded region, such as, what label is assigned to the occluded region, *shadow* or *background*. This kind of information is valuable for many multi-view applications.

The remainder of this paper is organized as follows. Section 2 introduces our segmentation framework. Section 3 describes the detailed implementation of the proposed method. Section 4 demonstrates the effectiveness of the proposed method using experiments. Finally, Section 5 concludes our work.

2 Segmentation Based on Homography-Correspondence Pairs

2.1 Problem Setting

In this paper, the following conditions are assumed in our segmentation problem.

- A scene is captured by a static calibrated binocular camera system.
- The background of the scene is modeled as a pixel-wise Gaussian distribution.

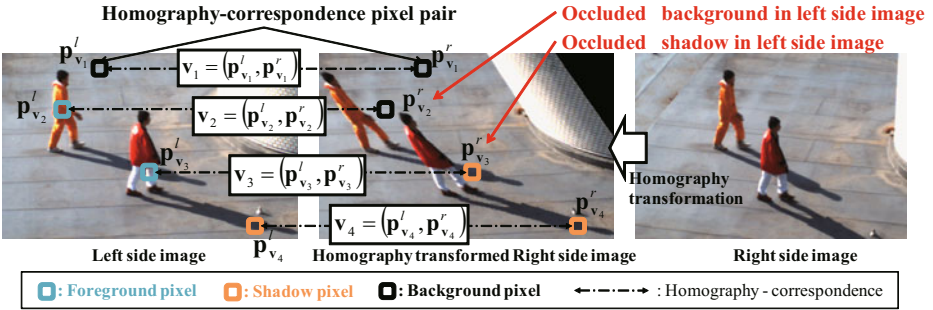


Fig. 1. Homography-correspondence pair

- An object in the foreground stands on the ground plane and its shadow appears on the ground plane.

Our goal is to segment the target region as *foreground* (“*F*”) or *shadow* (“*S*”) or *background* (“*B*”), that is to say, to assign one of the three labels “*F*”, “*S*”, or “*B*” to each pixel in both side images in a conformal manner. Note that “*S*” and “*B*” lie on the ground plane while “*F*” stands on the ground plane.

2.2 Asymmetric Treatment of Homography Constraint

Let us consider the homography-correspondence pair on the ground plane in the binocular camera. According to the homography constraint, if a pixel belongs to the ground plane on one side image, the color of the pixel is strictly consistent with that of the homography-correspondence pixel in the other side image under the condition that ideally any standing object does not exist on the ground plane. This is a very useful property to distinguish the standing objects on the ground plane from the ground plane objects. Many object detection techniques, for example, obstacle detection [12], shadow detection [13] [14] [15], have been proposed based on this property.

In segmentation problems, this property is also useful when assigning a label to each pixel. Some examples of the homography-correspondence pairs are shown in Fig. 1. First, suppose that the left side image is a base image to be segmented. Because v_1 and v_4 have similar colors between each correspondence pixel, $p_{v_1}^l$ and $p_{v_4}^l$ are labeled as “*S*” or “*B*” in the left side image. On the other hand, because the pixel pairs v_2 and v_3 have different colors between each pair of correspondence pixels, $p_{v_2}^l$ and $p_{v_3}^l$ are labeled as “*F*” in the left side image. Next, supposed that the right side image is a base image to be segmented in turn, pixel pairs $p_{v_1}^r$ and $p_{v_4}^r$ are labeled as “*S*” or “*B*”, and the pixel pairs $p_{v_2}^r$ and $p_{v_3}^r$ are labeled as “*F*” in the right side image in the same way. The true labels of $p_{v_2}^r$ and $p_{v_3}^r$ are, however, not “*F*” but “*B*” and “*S*”.

This mislabeling often arises in cases where a pixel belongs to the ground plane in one side image and where the corresponding pixel’s ground plane point in the other side image is occluded by a foreground object as shown in this example.

Table 1. The pair-wise label sets for a homography-correspondence pair

Left-side label	Right-side label		
	<i>F</i>	<i>S</i>	<i>B</i>
<i>F</i>	<i>FF</i>	<i>FS</i>	<i>FB</i>
<i>S</i>	<i>SF</i>	<i>SS</i>	-(prohibited)
<i>B</i>	<i>BF</i>	-(prohibited)	<i>BB</i>

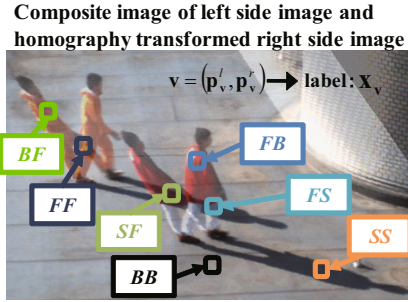


Fig. 2. Labeling examples

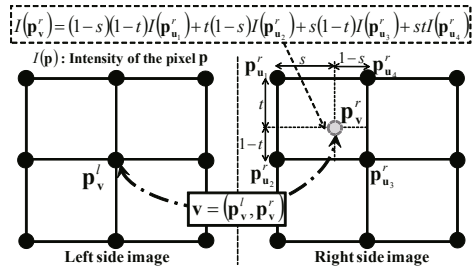


Fig. 3. Homography-correspondence detail

Therefore, the existing asymmetric homography-based approaches suffer from the mislabeling due to occlusion.

2.3 Symmetric Approach Based Homography-Correspondence Pair

In our framework, the homography-correspondence is treated symmetrically to cope with the occluded regions and to segment them correctly.

Taking the occlusion relationship into consideration, the labeling strategy is as follows. If the pixels are labeled “*S*” or “*B*” in one side image, their homography-correspondence pixels in the other side image are given either the same label (not the occluded case) or “*F*” (the occluded case). If the pixels in one side image are labeled “*F*”, their homography-correspondence pixels in the other side image are possibly labeled “*F*”, “*S*”, or “*B*”, because the standing object is not constrained by homography. From this observation, the possible pair-wise label for the homography-correspondence pair are defined in Table 1. Note that the pair-wise label “*SB*” and “*BS*” are prohibited because it is not possible for a ground plane object to occlude another ground plane object. Thus our segmentation problem is regarded as a *multi-labeling problem for homography-correspondence pair pixels*, and the labeling results provide all the relationships between homography-correspondence pair of pixels. For example, the label “*FS*” means the foreground occludes the shadow in the left side image, and also means the shadow in the right side image is occluded by the foreground in the left side image. Example of pair-wise labeling are shown in Fig. 2.

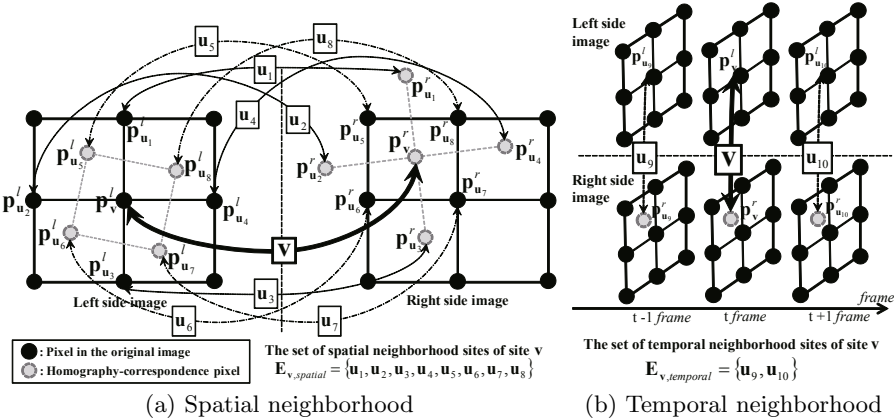


Fig. 4. Spatio-temporal neighborhood sites for smoothness term

2.4 Problem Formulation

We formulate the pair-wise multi-labeling problem in a framework that minimizes energy. Let us define the site $\mathbf{v} = (\mathbf{p}_\mathbf{v}^l, \mathbf{p}_\mathbf{v}^r)$ which represents a homography-correspondence pair as described in the previous subsection. Then, the label set is defined as,

$$\mathbf{L} = \{FF, FS, FB, SF, SS, BF, BB\}, \tag{1}$$

and the label assigned to a site \mathbf{v} as $\mathbf{x}_\mathbf{v} \in \mathbf{L}$. Then our goal is to assign each site \mathbf{v} a label $\mathbf{x}_\mathbf{v}$ from the set \mathbf{L} . Generally, this problem is formulated in an energy minimization framework as follows,

$$E(\mathbf{x}) = w_g \sum_{\mathbf{v} \in \mathbf{V}} \mathbf{g}(\mathbf{x}_\mathbf{v}) + w_h \sum_{(\mathbf{u}, \mathbf{v}) \in \mathbf{E}} \mathbf{h}(\mathbf{x}_\mathbf{u}, \mathbf{x}_\mathbf{v}) \tag{2}$$

where the first and the second terms are data and smoothness terms, w_g and w_h are the weights of each term, \mathbf{x} is a configuration (label combination), \mathbf{V} is a set of all sites, and \mathbf{E} is all the combinations of the neighborhood sites. This energy function is minimized via graph-cut algorithms such as the α -expansion or α - β swap algorithms [18].

Note that, the homography-correspondence positions are calculated using sub-pixel order and the color of the sub-pixel position is spatially interpolated by their 4-neighborhood pixels as shown in Fig. 3. In addition, as shown in Fig. 4, we consider 10-neighborhood sites in a spatio-temporal 3D domain composed of spatial 8-neighborhood, and temporal 2-neighborhood sites.

3 Implementation

3.1 Seed Generation

Given background subtraction regions as potential regions of shadow and foreground, the foreground seed is provided as the union of the following two regions;

one is the intersection of the potential region and background region projected by homography from the other image, and the other is the region which has a largely different color direction from the background one. Then, the shadow seed is decided based on homography consistency and color-based shadow likelihood (see Chapter 3.2 for detail).

3.2 Data Term

The data term is defined by the log of the likelihood as,

$$\mathbf{g}(\mathbf{x}_v) = -\log\left(P(\mathbf{x}_v|\mathbf{c}(\mathbf{v}))\right) = -\log\left(\frac{P(\mathbf{c}(\mathbf{v})|\mathbf{x}_v)P(\mathbf{x}_v)}{\sum_{l_i \in \mathbf{L}} P(\mathbf{c}(\mathbf{v})|\mathbf{x}_v=l_i)P(\mathbf{x}_v=l_i)}\right), \quad (3)$$

where $P()$ is probability and $\mathbf{c}(\mathbf{v})$ is a six dimensional color vector at site \mathbf{v} composed of a pair of RGB vectors in each image as where $P()$ is probability and $\mathbf{c}(\mathbf{v})$ is a six dimensional color vector at site \mathbf{v} composed of a pair of RGB vectors in each image as $\mathbf{c}(\mathbf{v}) = [\mathbf{c}(\mathbf{p}_v^l), \mathbf{c}(\mathbf{p}_v^r)]^T$, and $\mathbf{c}(\mathbf{p})$ is color vector at pixel \mathbf{p} . Then the pair-wise color observation model $P(\mathbf{c}(\mathbf{v})|\mathbf{x}_v)$ is decomposed into $\prod_i P(\mathbf{c}(\mathbf{p}_v^i)|\mathbf{x}_v^i)$, where \mathbf{x}_v^i is the one side label and i ($i = l, r$) is the camera identifier.

Foreground model. The foreground color is approximated by a pixel-wise GMM which is trained by k-means clustering from *foreground seed* pixels, and the foreground observation model is expressed as,

$$P(\mathbf{c}(\mathbf{p}_v^i)|\mathbf{x}_v^i = F) = \mathcal{N}(\mathbf{c}_f^{k*}, \sum_f^{k*}) \quad (4)$$

$$k^* = \arg \min_k \left((\mathbf{c}(\mathbf{p}_v^i) - \mathbf{c}_f^k)^T \sum_f^{k-1} (\mathbf{c}(\mathbf{p}_v^i) - \mathbf{c}_f^k) \right), \quad (5)$$

where \mathbf{c}_f^k and \sum_f^k are a mean vector and a covariance matrix of the k th cluster, and \mathcal{N} is the Gaussian distribution.

Shadow-Background model. First, a linear color transformation matrix from the background color to the shadow color is estimated from the *shadow seed* colors and their modeled background colors. This matrix is modeled as following a finite-dimensional linear model [19],

$$\mathbf{c}_s(\mathbf{p}) = \mathbf{A}\tilde{\mathbf{c}}_{bg}(\mathbf{p}), \quad (6)$$

where \mathbf{c}_s is a color vector of a shadow seed, $\tilde{\mathbf{c}}_{bg}$ is an extended color vector of a modeled background, $\tilde{\mathbf{c}}_{bg} = [\mathbf{c}_{bg}^T, 1]$, and \mathbf{A} is a 3 by 4 shadow transformation matrix. Then, the color transformation matrix \mathbf{A} is obtained by minimizing the following objective function S ,

$$\mathbf{e}(\mathbf{p}) = \mathbf{A}\tilde{\mathbf{c}}_{bg}(\mathbf{p}) - \mathbf{c}_s(\mathbf{p}) \quad (7)$$

$$S = \sum_{\mathbf{p} \in \mathbf{P}_s} \mathbf{e}(\mathbf{p})^T \left(\sum_{bg}(\mathbf{p}) \right)^{-1} \mathbf{e}(\mathbf{p}), \quad (8)$$

where d_e^i is an edge intensity criteria given by,

$$d_e^i(\mathbf{x}_v, \mathbf{x}_u) = \frac{\|\mathbf{c}(\mathbf{p}_v^i) - \mathbf{c}(\mathbf{p}_u^i)\|^2}{\|\mathbf{c}(\mathbf{p}_v^i) + \mathbf{c}(\mathbf{p}_u^i)\|^2 + \epsilon}, \tag{16}$$

where κ and ϵ are coefficients for this term.

4 Experiments

4.1 Data Set and Parameters

We carried out experiments using sequences of people walking outdoors. Table 2 shows the details of the data set. Every sequence contains some men or women with strong shadows. A total of 3 images were provided for graph-cut segmentation in a block. Note that in some figures in this section, the results of the experimental images are trimmed around the segmentation target region because page space is limited.

In these experiments, the data terms were spatially smoothed in response to the magnitude of the edge pixels. Because the pixel color is quite variable, and it is unstable near the edge, the reliability of the data terms is very low for such pixels. The segmentation process was done iteratively, and there were 2 iterations. The parameters of the proposed method were experimentally set at $w_g = 3.0$, $w_h = 0.3$, $\kappa = 4.0$, and $\epsilon = 10^{-7}$. Initially the prior of each label is set as follows: $P(FB) = P(BF) = P(SS) = 0.16$, $P(FS) = P(SF) = P(FF) = 0.14$, $P(BB) = 0.1$. In addition, the distribution number of GMM was set at 6 for *SeqA* and at 10 for *SeqB* and *SeqC*. We adopted the α - β swap algorithm [18] to minimize our energy function Eq. (2).

Table 2. Data set for experiments

Sequence set	Image size	Image number	Frame rate
<i>SeqA</i>	640×480	32	30 fps
<i>SeqB</i>	620×280	12	9 fps
<i>SeqC</i>	620×280	24	9 fps

4.2 Benchmark

We compared the segmentation performance of the following three approaches,

- *Color*: the color-based method and its implementation is as follows. First we generate the *foreground seed* and *shadow seed* based on the shadow color properties [5] [7]. Then we label each pixel in one side image as “F”, “S”, or “B”.
- *Color + Homography (asymmetric)*: the method integrating color and homography, and its implementation, follows. The seed generation process is the same as the proposed method, and the energy of the color similarities between the homography-correspondence pixels are integrated into a data term as *homography data term*. Then we label each pixel in the same way as the color-based method.
- *Color + Homography (symmetric)*: the proposed method.

Note that while the *Color + Homography (symmetric)* is a binocular-based symmetric framework, *Color* and *Color + Homography (asymmetric)* are not, so they are implemented as a “multi-labeling problem for a pixel” in each side of the image.

4.3 Results

First, the multi-labeling results of the proposed method for each data set are shown in Fig. 6, Fig. 7 and Fig. 8. In each result, the labeling results are good even for the occlusions.

Second, the performance comparison results for the ground truth which is created manually are shown in Fig. 9. In this figure, we can see that *Color* tends to fail at the pixels whose colors are quite similar to the shadow color. In situations where there is a black school bag (Fig. 9(a)) or a black-haired human head (Fig. 9(b)), this region is initially mislabeled *shadow seed*. Furthermore, it is a problem of the color-based approach that the pixels whose colors are similar to the shadow cannot be identified in the foreground GMM model, because such pixels are poorly labeled as a *foreground seed*, so the foreground data terms of such pixels are very low, and as a result, these are mislabeled as shadows. This is inevitable for the color-based approach.

On the other hand, for the results of the *Color + Homography (asymmetric)* and *Color + Homography (symmetric)* approaches, such pixels are correctly labeled as foreground. This is because such pixels are initially labeled as foreground seeds by the homography constraints, and the foreground GMM model includes such color information, and foreground data terms are high. As for the results of *Color + Homography (asymmetric)*, however, we can see the occlusion problem as described in Section 2. In contrast, *Color + Homography (symmetric)* segments them correctly.

The quantitative performance comparisons are shown in Table 3. The performance of each method is evaluated by *F-measure*, which is defined as,

$$F = \frac{2PR}{P + R}, \quad (17)$$

where F is *F-measure*, and P and R are *precision* and *recall*.

In the tables, we see that the *Color + Homography (symmetric)* approach totally outperforms the other methods. For the *SeqA*, there is little difference between the *Color + Homography (asymmetric)* and the *Color + Homography (symmetric)* approaches. This is because the color data and homography data terms are well balanced in this sequence.

Table 3. Quantitative evaluation results

Method	<i>SeqA</i>		<i>SeqB</i>		<i>SeqC</i>	
	<i>f</i>	<i>s</i>	<i>f</i>	<i>s</i>	<i>f</i>	<i>s</i>
<i>Color</i>	0.890	0.863	0.825	0.747	0.817	0.776
<i>Color + Homography (asymmetric)</i>	0.938	0.923	0.904	0.824	0.874	0.824
<i>Color + Homography (symmetric)</i>	0.940	0.900	0.919	0.858	0.899	0.864

f: foreground, *s*: shadow

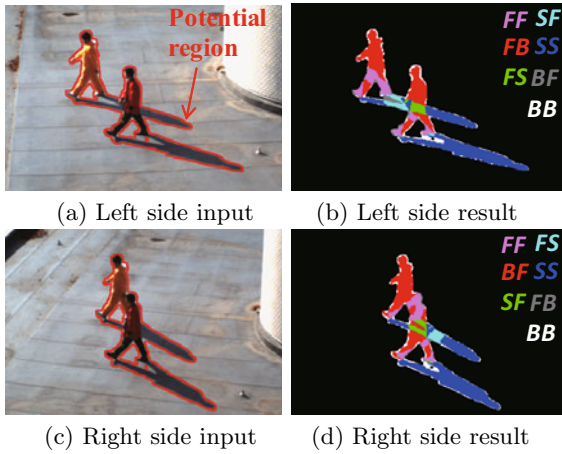


Fig. 6. Input and segmentation results of *SeqA*

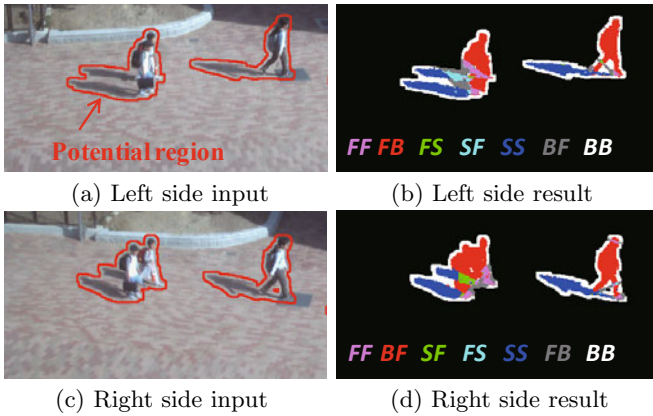


Fig. 7. Input and segmentation results of *SeqB*

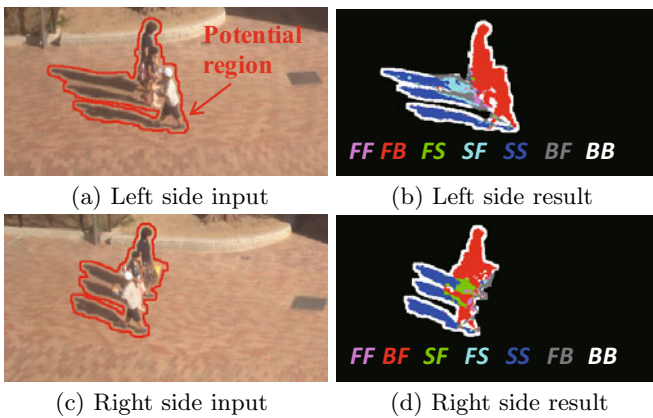


Fig. 8. Input and segmentation results of *SeqC*

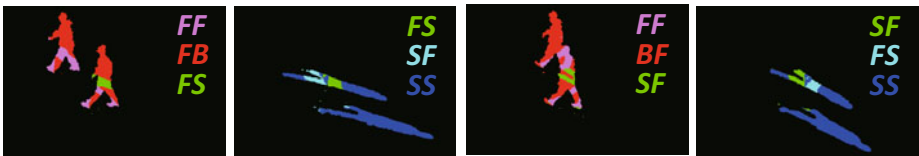
	<i>Seed</i>	Segmentation result (left side image)	
		foreground	shadow
<i>Color</i>			
<i>Color + Homography (asymmetric)</i>			
<i>Color + Homography (symmetric)</i>			

(a) *Seq B*

	<i>Seed</i>	Segmentation result (left side image)	
		foreground	shadow
<i>Color</i>			
<i>Color + Homography (asymmetric)</i>			
<i>Color + Homography (symmetric)</i>			

(b) *Seq C*

Fig. 9. Comparison Results



(a) Left side result

(b) Right side result

Fig. 10. Extracted foreground and a whole shadow including occluded shadow for *SeqA*

4.4 Discussions

Effective use of extracted shadow. By making effective use of extracted shadow, our approach can obtain consistent labeling as well as information as to

whether the occluded region belongs to the *shadow* or *background*. This means that we can get additional scene information. For example, because a whole shadow silhouette including the occluded shadow, can be seen as another projection from the viewpoint of a light source, we can say that one more different-view of the whole silhouette of the target foreground objects is extracted as shown in Fig. 10. This is quite valuable for many computer vision applications, especially silhouette based applications, like gait recognition, gesture recognition, 3D reconstruction by shape from silhouettes and so on. As for gait recognition, it is reported in [20] that the different views of silhouettes improve recognition, and more, shadow-based gait recognition scheme is proposed in [21].

In addition, homography-based object localization techniques have been proposed [15], where the position of the object is localized by estimating the intersection point of the object region and the shadow region. Hence, if the occluded shadow region is also extracted by the proposed method, the object localization accuracy is improved.

Extension to more complex scene or moving platform. Although the assumption that the shadow appears on the ground plane may seem to be a heavy constraint, our method can be extended to more complex scenes by modeling scenes as piecewise facets and by calibrating the homography for each facet.

Furthermore, our method can be applied to a mobile platform such as a vehicle binocular video system, and an intelligent robot with a combination of state of the art dynamic background modeling, ego-motion, and image stabilizing techniques. For example, we can acquire a background model for each frame of the image sequence by using dynamic background modeling, and we can calibrate the geometric relationship between the binocular camera system and the target plane by using ego-motion and image stabilizing techniques.

5 Conclusions

In this paper, we propose a homography-correspondence pair based segmentation framework. We treat homography-correspondence pairs symmetrically, and formulate the segmentation problem as a multi-labeling problem for a homography-correspondence pair to explicitly take the occlusion relationship into account. Then we obtain the segmentation result by minimizing the energy function via the α - β swap algorithm. In our experiments, it turns out that the segmentation results of the proposed method outperform the existing color-based and asymmetric homography-based methods.

Acknowledgement. This work was supported by Grant-in-Aid for Scientific Research(S) 21220003.

References

1. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, pp. 246–252 (1999)

2. Choi, J., Jun, Y., Choi, J.Y.: Adaptive shadow estimator for removing shadow of moving object. *Computer Vision and Image Understanding* 114, 1017–1029 (2010)
3. Cucchiara, R., Grana, C., Piccardi, M., Prati, A.: Detection moving objects, ghosts, and shadows in video streams. *IEEE Trans. Pattern Analysis and Machine Intelligence* 25, 1337–1342 (2003)
4. Horprasert, T., Harwood, D., Davis, L.S.: A robust background subtraction and shadow detection. In: *Proc. of the 4th Asian Conference on Computer Vision*, pp. 983–988 (2000)
5. Huang, J.B., Chen, C.S.: Moving cast shadows detection using physics-based features. In: *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 2310–2317 (2009)
6. Kakuta, T., Vinh, L.B., Kawakami, R., Oishi, T., Ikeuchi, K.: Detection of moving objects and cast shadows using a spherical vision camera for outdoor mixed reality. In: *Proc. on 15th ACM Symposium on Virtual Reality Software and Technology*, pp. 219–222 (2008)
7. Porikli, F., Thornton, J.: Shadow flow: A recursive method to learn moving cast shadows. In: *Proc. IEEE Int. Conf. on Computer Vision.*, vol. 1, pp. 891–898 (2005)
8. Prati, A., Mikic, I., Trivedi, M.M., Cucchiara, R.: Detecting moving shadows: Algorithms and evaluation. *IEEE Trans. Pattern Analysis and Machine Intelligence* 25, 918–923 (2003)
9. Tanaka, T., Shimada, A., Arita, D., Taniguchi, R.-i.: Non-parametric background and shadow modeling for object detection. In: Yagi, Y., Kang, S.B., Kweon, I.S., Zha, H. (eds.) *ACCV 2007, Part I. LNCS*, vol. 4843, pp. 159–168. Springer, Heidelberg (2007)
10. Gordon, G., Darrell, T., Harville, M., Woodfill, J.: Background estimation and removal based on range and color. In: *Proc. of the 18th Int. Conf. on Pattern Recognition*, pp. 459–464 (1999)
11. Madsen, C.B., Moeslund, T.B., Pal, A., Balasubramanian, S.: Shadow detection in dynamic scenes using dense stereo information and an outdoor illumination model. In: *Proc. on the DAGM, Workshop on Dynamic 3D Imaging*, pp. 110–125 (2009)
12. Batavia, P.H., Singh, S.: Obstacle detection using adaptive color segmentation and color stereo homography. In: *Proc. of IEEE Int. Conf. on Robotics and Automation.*, vol. 1, pp. 705–710 (2001)
13. Hamid, R., KrishanKumar, R., Grundmann, M., Kim, K., Essa, I., Hodgins, J.: Player localization using multiple static cameras for sports visualization. In: *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 731–738 (2010)
14. Jeong, K., Jaynes, C.: Moving shadow detection using a combined geometric and color classification approach. In: *Proc IEEE Workshop on Motion and Video Computing 2005*, vol. 2, pp. 36–43 (2005)
15. Kasuya, N., Kitahara, I., Kameda, Y., Ohta, Y.: Robust trajectory estimation of soccer players by using two cameras. In: *Proc. of the 19th Int. Conf. on Pattern Recognition*, pp. 1–4 (2008)
16. Kolmogorov, V., Zabih, R.: Computing visual correspondence with occlusions via graph cuts. In: *Proc. IEEE Int. Conf. on Computer Vision*, pp. 508–515 (2001)
17. Kolmogorov, V., Zabih, R.: Graph cut algorithms for binocular stereo with occlusions. In: *Mathematical Models in Computer Vision: The Handbook*, pp. 423–438 (2005)
18. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Analysis and Machine Intelligence* 23, 1222–1239 (2001)

19. Marimont, D.H., Wandell, B.A.: Linear models for surface and illumination spectra. *Journal of the Optical Society of America*, 1905–1913 (1992)
20. Sugiura, K., Makihara, Y., Yagi, Y.: Gait identification based on multi-view observations using omnidirectional camera. In: Yagi, Y., Kang, S.B., Kweon, I.S., Zha, H. (eds.) *ACCV 2007, Part I. LNCS*, vol. 4843, pp. 452–461. Springer, Heidelberg (2007)
21. Iwashita, Y., Stoica, A.: Gait recognition using shadow analysis. In: *2009 Bio-inspired Learning and Intelligent Systems for Security*, pp. 26–31 (2009)

Author Index

- Abdala, Daniel Duarte IV-373
Abe, Daisuke IV-565
Achtenberg, Albert IV-141
Ackermann, Hanno II-464
Agapito, Lourdes IV-460
Ahn, Jae Hyun IV-513
Ahuja, Narendra IV-501
Ai, Haizhou II-174, II-683, III-171
Akbas, Emre IV-501
Alexander, Andrew L. I-65
An, Yaozu II-282
Ancuti, Codruta O. I-79, II-501
Ancuti, Cosmin I-79, II-501
Argyros, Antonis A. III-744
Arnaud, Elise IV-361
Åström, Kalle IV-255
Atasoy, Selen II-41
Azuma, Takeo III-641
- Babari, Raouf IV-243
Badino, Hernán III-679
Badrinath, G.S. II-321
Bai, Li II-709
Ballan, Luca III-613
Barlaud, Michel III-67
Barnes, Nick I-176, IV-269, IV-410
Bartoli, Adrien III-52
Bekaert, Philippe I-79, II-501
Belhumeur, Peter N. I-39
Bennamoun, Mohammed III-199,
IV-115
Ben-Shahar, Ohad II-346
Ben-Yosef, Guy II-346
Binder, Alexander III-95
Bischof, Horst I-397, II-566
Bishop, Tom E. II-186
Biswas, Sujoy Kumar I-244
Bonde, Ujwal D. IV-228
Bowden, Richard I-256, IV-525
Boyer, Edmond IV-592
Brémond, Roland IV-243
Briassouli, Alexia I-149
Brocklehurst, Kyle III-329, IV-422
Brunet, Florent III-52
- Bu, Jiajun III-436
Bujnak, Martin I-11, II-216
Burschka, Darius I-135, IV-474
Byröd, Martin IV-255
- Carlsson, Stefan II-1
Cha, Joonhyuk IV-486
Chan, Kwok-Ping IV-51
Chen, Chia-Ping I-355
Chen, Chun III-436
Chen, Chu-Song I-355
Chen, Duowen I-283
Chen, Kai III-121
Chen, Tingwang II-400
Chen, Wei II-67
Chen, Yan Qiu IV-435
Chen, Yen-Wei III-511, IV-39, IV-165
Chen, Yi-Ling III-535
Chen, Zhihu II-137
Chi, Yu-Tseh II-268
Chia, Liang-Tien II-515
Chin, Tat-Jun IV-553
Cho, Nam Ik IV-513
Chu, Wen-Sheng I-355
Chum, Ondřej IV-347
Chung, Ronald H-Y. IV-690
Chung, Sheng-Luen IV-90
Collins, Maxwell D. I-65
Collins, Robert T. III-329, IV-422
Cootes, Tim F. I-1
Cosker, Darren IV-189
Cowan, Brett R. IV-385
Cree, Michael J. IV-397
Cremers, Daniel I-53
- Dai, Qionghai II-412
Dai, Zhenwen II-137
Danielsson, Oscar II-1
Davis, James W. II-580
de Bruijne, Marleen II-160
Declercq, Arnaud III-422
Deguchi, Koichiro IV-565
De la Torre, Fernando III-679
Delaunoy, Amaël I-39, II-55
Denzler, Joachim II-489

- De Smet, Michaël III-276
 Detry, Renaud III-572
 Dickinson, Sven I-369, IV-539
 Dikmen, Mert IV-501
 Ding, Jianwei II-82
 Di Stefano, Luigi III-653
 Dorothy, Monekosso I-439
 Dorrington, Adrian A. IV-397
 Duan, Genquan II-683
 Dumont, Éric IV-243
- El Ghouli, Aymen II-647
 Ellis, Liam IV-525
 Eng, How-Lung I-439
 Er, Guihua II-412
- Fan, Yong IV-606
 Fang, Tianhong II-633
 Favaro, Paolo I-425, II-186
 Felsberg, Michael IV-525
 Feng, Jufu III-213, III-343
 Feng, Yaokai I-296
 Feragen, Aasa II-160
 Feuerstein, Marco III-409
 Fieguth, Paul I-383
 Förstner, Wolfgang II-619
 Franco, Jean-Sébastien III-599
 Franek, Lucas II-697, IV-373
 Fu, Yun II-660
 Fujimura, Ikko I-296
 Fujiyoshi, Hironobu IV-25
 Fukuda, Hisato IV-127
 Fukui, Kazuhiro IV-580
 Furukawa, Ryo IV-127
- Ganesh, Arvind III-314, III-703
 Gao, Changxin III-133
 Gao, Yan IV-153
 Garg, Ravi IV-460
 Geiger, Andreas I-25
 Georgiou, Andreas II-41
 Ghahramani, M. II-388
 Gilbert, Andrew I-256
 Godbaz, John P. IV-397
 Gong, Haifeng II-254
 Gong, Shaogang I-161, II-293, II-527
 Gopalakrishnan, Viswanath II-15,
 III-732
 Grabner, Helmut I-200
 Gu, Congcong III-121
- Gu, Steve I-271
 Guan, Haibing III-121
 Guo, Yimo III-185
 Gupta, Phalguni II-321
- Hall, Peter IV-189
 Han, Shuai I-323
 Hao, Zhihui IV-269
 Hartley, Richard II-554, III-52,
 IV-177, IV-281
 Hauberg, Søren III-758
 Hautière, Nicolas IV-243
 He, Hangen III-27
 He, Yonggang III-133
 Helmer, Scott I-464
 Hendel, Avishai III-448
 Heo, Yong Seok IV-486
 Hermans, Chris I-79, II-501
 Ho, Jeffrey II-268
 Horaud, Radu IV-592
 Hospedales, Timothy M. II-293
 Hou, Xiaodi III-225
 Hsu, Gee-Sern IV-90
 Hu, Die II-672
 Hu, Tingbo III-27
 Hu, Weiming II-594, III-691, IV-630
 Hu, Yiqun II-15, II-515, III-732
 Huang, Jia-Bin III-497
 Huang, Kaiqi II-67, II-82, II-542
 Huang, Thomas S. IV-501
 Huang, Xincheng IV-281
 Huang, Yongzhen II-542
 Hung, Dao Huu IV-90
- Igarashi, Yosuke IV-580
 Ikemura, Sho IV-25
 Iketani, Akihiko III-109
 Imagawa, Taro III-641
 Iwama, Haruyuki IV-702
- Jankó, Zsolt II-55
 Jeon, Moongu III-718
 Jermyn, Ian H. II-647
 Ji, Xiangyang II-412
 Jia, Ke III-586
 Jia, Yunde II-254
 Jiang, Hao I-228
 Jiang, Mingyang III-213, III-343
 Jiang, Xiaoyi II-697, IV-373
 Jung, Soon Ki I-478

- Kakadiaris, Ioannis A. II-633
 Kale, Amit IV-592
 Kambhamettu, Chandra III-82, III-483,
 III-627
 Kanatani, Kenichi II-242
 Kaneda, Kazufumi II-452, III-250
 Kang, Sing Bing I-350
 Kasturi, Rangachar II-308
 Kawabata, Satoshi III-523
 Kawai, Yoshihiro III-523
 Kawanabe, Motoaki III-95
 Kawano, Hiroki I-296
 Kawasaki, Hiroshi IV-127
 Kemmler, Michael II-489
 Khan, R. Nazim III-199
 Kikutsugi, Yuta III-250
 Kim, Du Yong III-718
 Kim, Hee-Dong IV-1
 Kim, Hyunwoo IV-333
 Kim, Jaewon I-336
 Kim, Seong-Dae IV-1
 Kim, Sujung IV-1
 Kim, Tae-Kyun IV-228
 Kim, Wook-Joong IV-1
 Kise, Koichi IV-64
 Kitasaka, Takayuki III-409
 Klinkigt, Martin IV-64
 Kompatsiaris, Ioannis I-149
 Kopp, Lars IV-255
 Kuang, Gangyao I-383
 Kuang, Yubin IV-255
 Kuk, Jung Gap IV-513
 Kukulova, Zuzana I-11, II-216
 Kulkarni, Kaustubh IV-592
 Kwon, Dongjin I-121
 Kyriazis, Nikolaos III-744

 Lai, Shang-Hong III-535
 Lam, Antony III-157
 Lao, Shihong II-174, II-683, III-171
 Lauze, Francois II-160
 Lee, Kyong Joon I-121
 Lee, Kyoung Mu IV-486
 Lee, Sang Uk I-121, IV-486
 Lee, Sukhan IV-333
 Lei, Yinjie IV-115
 Levinstein, Alex I-369
 Li, Bing II-594, III-691, IV-630
 Li, Bo IV-385
 Li, Chuan IV-189
 Li, Chunxiao III-213, III-343
 Li, Fajie IV-641
 Li, Hongdong II-554, IV-177
 Li, Hongming IV-606
 Li, Jian II-293
 Li, Li III-691
 Li, Min II-67, II-82
 Li, Sikun III-471
 Li, Wei II-594, IV-630
 Li, Xi I-214
 Li, Yiqun IV-153
 Li, Zhidong II-606, III-145
 Liang, Xiao III-314
 Little, James J. I-464
 Liu, Jing III-239
 Liu, Jingchen IV-102
 Liu, Li I-383
 Liu, Miaomiao II-137
 Liu, Nianjun III-586
 Liu, Wei IV-115
 Liu, Wenyu III-382
 Liu, Yanxi III-329, IV-102, IV-422
 Liu, Yong III-679
 Liu, Yonghuai II-27
 Liu, Yuncai II-660
 Lladó, X. III-15
 Lo, Pechin II-160
 Lovell, Brian C. III-547
 Lowe, David G. I-464
 Loy, Chen Change I-161
 Lu, Feng II-412
 Lu, Guojun IV-449
 Lu, Hanqing III-239
 Lu, Huchuan III-511, IV-39, IV-165
 Lu, Shipeng IV-165
 Lu, Yao II-282
 Lu, Yifan II-554, IV-177
 Lu, Zhaojin IV-333
 Luo, Guan IV-630
 Luó, Xióngbiāo III-409
 Luo, Ye III-396

 Ma, Songde III-239
 Ma, Yi III-314, III-703
 MacDonald, Bruce A. II-334
 MacNish, Cara III-199
 Macrini, Diego IV-539
 Mahalingam, Gayathri III-82
 Makihara, Yasushi I-107, II-440,
 III-667, IV-202, IV-702

- Makris, Dimitrios III-262
 Malgouyres, Remy III-52
 Mannami, Hidetoshi II-440
 Martin, Ralph R. II-27
 Matas, Jiří IV-347
 Matas, Jiri III-770
 Mateus, Diana II-41
 Matsushita, Yasuyuki I-336, III-703
 Maturana, Daniel IV-618
 Mauthner, Thomas II-566
 McCarthy, Chris IV-410
 Meger, David I-464
 Mehdizadeh, Maryam III-199
 Mery, Domingo IV-618
 Middleton, Lee I-200
 Moon, Youngsu IV-486
 Mori, Atsushi I-107
 Mori, Kensaku III-409
 Muja, Marius I-464
 Mukaigawa, Yasuhiro I-336, III-667
 Mukherjee, Dipti Prasad I-244
 Mukherjee, Snehasis I-244
 Müller, Christina III-95
- Nagahara, Hajime III-667, IV-216
 Nakamura, Ryo II-109
 Nakamura, Takayuki IV-653
 Navab, Nassir II-41, III-52
 Neumann, Lukas III-770
 Nguyen, Hieu V. II-709
 Nguyen, Tan Dat IV-665
 Nielsen, Frank III-67
 Nielsen, Mads II-160
 Niitsuma, Hirotaka II-242
 Nock, Richard III-67
- Oikonomidis, Iasonas III-744
 Okabe, Takahiro I-93, I-323
 Okada, Yusuke III-641
 Okatani, Takayuki IV-565
 Okutomi, Masatoshi III-290, IV-76
 Okwechime, Dumebi I-256
 Ommert, Björn II-477
 Ong, Eng-Jon I-256
 Ortner, Mathias IV-361
 Orwell, James III-262
 Oskarsson, Magnus IV-255
 Oswald, Martin R. I-53
 Ota, Takahiro IV-653
- Paisitkriangkrai, Sakrapee III-460
 Pajdla, Tomas I-11, II-216
 Pan, ChunHong II-148, III-560
 Pan, Xiuxia IV-641
 Paparoditis, Nicolas IV-243
 Papazov, Chavdar I-135
 Park, Minwoo III-329, IV-422
 Park, Youngjin III-355
 Pedersen, Kim Steenstrup III-758
 Peleg, Shmuel III-448
 Peng, Xi I-283
 Perrier, Régis IV-361
 Piater, Justus III-422, III-572
 Pickup, David IV-189
 Pietikäinen, Matti III-185
 Piro, Paolo III-67
 Pirri, Fiora III-369
 Pizarro, Luis IV-460
 Pizzoli, Matia III-369
 Pock, Thomas I-397
 Pollefeys, Marc III-613
 Prados, Emmanuel I-39, II-55
 Provenzi, E. III-15
- Qi, Baojun III-27
- Rajan, Deepu II-15, II-515, III-732
 Ramakrishnan, Kalpatti R. IV-228
 Ranganath, Surendra IV-665
 Raskar, Ramesh I-336
 Ravichandran, Avinash I-425
 Ray, Nilanjan III-39
 Raytchev, Bisser II-452, III-250
 Reddy, Vikas III-547
 Reichl, Tobias III-409
 Remagnino, Paolo I-439
 Ren, Zhang I-176
 Ren, Zhixiang II-515
 Rodner, Erik II-489
 Rohith, MV III-627
 Rosenhahn, Bodo II-426, II-464
 Roser, Martin I-25
 Rosin, Paul L. II-27
 Roth, Peter M. II-566
 Rother, Carsten I-53
 Roy-Chowdhury, Amit K. III-157
 Rudi, Alessandro III-369
 Rudoy, Dmitry IV-307
 Rueckert, Daniel IV-460
 Ruepp, Oliver IV-474

- Sagawa, Ryusuke III-667
 Saha, Baidya Nath III-39
 Sahbi, Hichem I-214
 Sakaue, Fumihiko II-109
 Sala, Pablo IV-539
 Salti, Samuele III-653
 Salvi, J. III-15
 Sanderson, Conrad III-547
 Sang, Nong III-133
 Sanin, Andres III-547
 Sankaranarayanan, Karthik II-580
 Santner, Jakob I-397
 Šára, Radim I-450
 Sato, Imari I-93, I-323
 Sato, Jun II-109
 Sato, Yoichi I-93, I-323
 Savoye, Yann III-599
 Scheuermann, Björn II-426
 Semenovich, Dimitri I-490
 Senda, Shuji III-109
 Shah, Shishir K. II-230, II-633
 Shahiduzzaman, Mohammad IV-449
 Shang, Lifeng IV-51
 Shelton, Christian R. III-157
 Shen, Chunhua I-176, III-460,
 IV-269, IV-281
 Shi, Boxin III-703
 Shibata, Takashi III-109
 Shimada, Atsushi IV-216
 Shimano, Mihoko I-93
 Shin, Min-Gil IV-293
 Shin, Vladimir III-718
 Sigal, Leonid III-679
 Singh, Vikas I-65
 Sminchisescu, Cristian I-369
 Somanath, Gowri III-483
 Song, Li II-672
 Song, Ming IV-606
 Song, Mingli III-436
 Song, Ran II-27
 Soto, Álvaro IV-618
 Sowmya, Arcot I-490, II-606
 Stol, Karl A. II-334
 Sturm, Peter IV-127, IV-361
 Su, Hang III-302
 Su, Te-Feng III-535
 Su, Yanchao II-174
 Sugimoto, Shigeki IV-76
 Sung, Eric IV-11
 Suter, David IV-553
 Swadzba, Agnes II-201
 Sylwan, Sebastian I-189
 Szirányi, Tamás IV-321
 Szolgay, Dániel IV-321
 Tagawa, Seiichi I-336
 Takeda, Takahishi II-452
 Takemura, Yoshito II-452
 Tamaki, Toru II-452, III-250
 Tan, Tieniu II-67, II-82, II-542
 Tanaka, Masayuki III-290
 Tanaka, Shinji II-452
 Taneja, Aparna III-613
 Tang, Ming I-283
 Taniguchi, Rin-ichiro IV-216
 Tao, Dacheng III-436
 Teoh, E.K. II-388
 Thida, Myo I-439
 Thomas, Stephen J. II-334
 Tian, Qi III-239, III-396
 Tian, Yan III-679
 Timofte, Radu I-411
 Tomasi, Carlo I-271
 Tombari, Federico III-653
 Töppe, Eno I-53
 Tossavainen, Timo III-1
 Trung, Ngo Thanh III-667
 Tyleček, Radim I-450
 Uchida, Seiichi I-296
 Ugawa, Sanzo III-641
 Urtasun, Raquel I-25
 Vakili, Vida II-123
 Van Gool, Luc I-200, I-411, III-276
 Vega-Pons, Sandro IV-373
 Veksler, Olga II-123
 Velastin, Sergio A. III-262
 Veres, Galina I-200
 Vidal, René I-425
 Wachsmuth, Sven II-201
 Wada, Toshikazu IV-653
 Wagner, Jenny II-477
 Wang, Aiping III-471
 Wang, Bo IV-269
 Wang, Hanzi IV-630
 Wang, Jian-Gang IV-11
 Wang, Jinqiao III-239
 Wang, Lei II-554, III-586, IV-177

- Wang, LingFeng II-148, III-560
 Wang, Liwei III-213, III-343
 Wang, Nan III-171
 Wang, Peng I-176
 Wang, Qing II-374, II-400
 Wang, Shao-Chuan I-310
 Wang, Wei II-95, III-145
 Wang, Yang II-95, II-606, III-145
 Wang, Yongtian III-703
 Wang, Yu-Chiang Frank I-310
 Weinshall, Daphna III-448
 Willis, Phil IV-189
 Wojcikiewicz, Wojciech III-95
 Won, Kwang Hee I-478
 Wong, Hoi Sim IV-553
 Wong, Kwan-Yee K. II-137, IV-690
 Wong, Wilson IV-115
 Wu, HuaiYu III-560
 Wu, Lun III-703
 Wu, Ou II-594
 Wu, Tao III-27
 Wu, Xuqing II-230

 Xiang, Tao I-161, II-293, II-527
 Xiong, Weihua II-594
 Xu, Changsheng III-239
 Xu, Dan II-554, IV-177
 Xu, Jie II-95, II-606, III-145
 Xu, Jiong II-374
 Xu, Zhengguang III-185
 Xue, Ping III-396

 Yaegashi, Keita II-360
 Yagi, Yasushi I-107, I-336, II-440,
 III-667, IV-202, IV-702
 Yamaguchi, Takuma IV-127
 Yamashita, Takayoshi II-174
 Yan, Ziyue II-282
 Yanai, Keiji II-360
 Yang, Chih-Yuan III-497
 Yang, Ehwa III-718
 Yang, Fan IV-39
 Yang, Guang-Zhong II-41
 Yang, Hua III-302
 Yang, Jie II-374
 Yang, Jun II-95, II-606, III-145
 Yang, Ming-Hsuan II-268, III-497
 Yau, Wei-Yun IV-11
 Yau, W.Y. II-388
 Ye, Getian II-95

 Yin, Fei III-262
 Yoo, Suk I. III-355
 Yoon, Kuk-Jin IV-293
 Yoshida, Shigeto II-452
 Yoshimuta, Junki II-452
 Yoshinaga, Satoshi IV-216
 Young, Alistair A. IV-385
 Yu, Jin IV-553
 Yu, Zeyun II-148
 Yuan, Chunfeng III-691
 Yuan, Junsong III-396
 Yuk, Jacky S-C. IV-690
 Yun, Il Dong I-121

 Zappella, L. III-15
 Zeevi, Yehoshua Y. IV-141
 Zelnik-Manor, Lihi IV-307
 Zeng, Liang III-471
 Zeng, Zhihong II-633
 Zerubia, Josiane II-647
 Zhang, Bang II-606
 Zhang, Chunjie III-239
 Zhang, Dengsheng IV-449
 Zhang, Hong III-39
 Zhang, Jian III-460
 Zhang, Jing II-308
 Zhang, Liqing III-225
 Zhang, Luming III-436
 Zhang, Wenling III-511
 Zhang, Xiaoqin IV-630
 Zhang, Zhengdong III-314
 Zhao, Guoying III-185
 Zhao, Xu II-660
 Zhao, Youdong II-254
 Zheng, Hong I-176
 Zheng, Huicheng IV-677
 Zheng, Qi III-121
 Zheng, Shibao III-302
 Zheng, Wei-Shi II-527
 Zheng, Ying I-271
 Zheng, Yinqiang IV-76
 Zheng, Yongbin IV-281
 Zhi, Cheng II-672
 Zhou, Bolei III-225
 Zhou, Quan III-382
 Zhou, Yi III-121
 Zhou, Yihao IV-435
 Zhou, Zhuoli III-436
 Zhu, Yan II-660