Ron Kimmel
Reinhard Klette
Akihiro Sugimoto (Eds.)

LNCS 6494

Computer Vision – ACCV 2010

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

Springer

Lecture Notes in Computer Science 6494

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

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 III

Springer

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 978-3-642-19317-0
DOI 10.1007/978-3-642-19318-7

e-ISSN 1611-3349
e-ISBN 978-3-642-19318-7

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

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
Achard, Catherine
Ai, Haizhou
Aiger, Dror
Alahari, Karteek
Araguas, Gaston
Arica, Nafiz
Ariki, Yasuo
Arslan, Abdullah
Astroem, Kalle
August, Jonas
Aura Vese, Luminita
Azevedo-Marques, Paulo
Bagdanov, Andy
Bagon, Shai
Bai, Xiang
Baloch, Sajjad
Baltes, Jacky
Bao, Yufang
Bar, Leah
Barbu, Adrian
Barnes, Nick
Barron, John
Bartoli, Adrien
Baust, Maximilian
Ben Hamza, Abdessamad
BenAbdelkader, Chiraz
Ben-ari, Rami
Beng-Jin, AndrewTeoh
Benosman, Ryad
Berkels, Benjamin
Berthier, Michel
Bhattacharya, Bhargab
Biswas, Prabir
Bo, Liefeng
Boerdgen, Markus
Bors, Adrian
Boshra, Michael
Bouguila, Nizar
Boyer, Edmond
Bronstein, Michael
Bruhn, Andres
Buckley, Michael
Cai, Jinhai
Cai, Zhenjiang
Calderón, Jesús
Camastra, Francesco
Canavesio, Luisa
Cao, Xun
Carlo, Colombo
Carlsson, Stefan
Caspi, Yaron
Castellani, Umberto
Celik, Turgay
Cham, Tat-Jen
Chan, Antoni
Chandran, Sharat
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, Kourosh
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, Sudipta
So Kweon, In
Sommerlade, Eric
Song, Andy
Souvenir, Richard
Srivastava, Anuj
Staiano, Jacopo
Stein, Gideon
Stottinge, Julian
Strecha, Christoph
Strekalovskiy, 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, Suya
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
Collins, Toby
Compte, Benot
Cong, Yang
Das, Samarjit
Duan, Lixing
Fihl, Preben
Garro, Valeria
Geng, Bo
Gherardi, Riccardo
Giusti, Alessandro
Guo, Jing-Ming
Gupta, Vipin
Han, Long
Korchev, Dmitriy
Kulkarni, Kaustubh
Lewandowski, Michal
Li, Xin
Li, Zhu
Lin, Guo-Shiang
Lin, Wei-Yang
Liu, Damon Shing-Min
Liu, Dong
Luo, Ye
Magerand, Ludovic
Molineros, Jose
Rao, Shankar
Samir, Chafik
Sanchez-Riera, Jordy
Suryanarayana, Venkata
Tang, Sheng
Thota, Rahul
Toldo, Roberto
Tran, Du
Wang, Jingdong
Wu, Jun
Yang, Jianchao
Yang, Linjun
Yang, Kuiyuan
Yuan, Fei
Zhang, Guofeng
Zhuang, Jinfeng

ACCV2010 Best Paper Award Committee

Alfred M. Bruckstein	Technion, Israel Institute of Techonlogy, 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	*The International Journal of Computer Vision* (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 III

Posters on Day 2 of ACCV 2010

Approximate and SQP Two View Triangulation

Timo Tossavainen

Department of Media Technology,
Aalto University School of Science and Technology, Finland
Timo.Tossavainen@tkk.fi

Abstract. The two view triangulation problem with Gaussian errors, aka optimal triangulation, has an optimal solution that requires finding the roots of a 6th degree polynomial. This is computationally quite demanding for a basic building block of many reconstruction algorithms. We consider two faster triangulation methods. The first is a closed form approximate solution that comes with intuitive and tight error bounds that also describe cases where the optimal method is needed. The second is an iterative method based on local sequential quadratic programming (SQP). In simulations, triangulation errors of the approximate method are on par with the optimal method in most cases of practical interest and the triangulation errors of the SQP method are on par with the optimal method in practically all cases. The SQP method is faster of the two and about two orders of magnitude faster than the optimal method.

1 Introduction

Triangulation, finding the point in space that projects to given target points in images of known cameras, is a fundamental operation in 3D reconstruction. Usually the target points are inaccurate measurements. Because of this inaccuracy, the backprojection rays from the cameras will not intersect exactly and the point has to be chosen according to some criterion. There are many variations on the theme, such as triangulation with different error metrics [1], triangulation for special camera configurations, such as the three view case [2], and triangulation by tensor approximations [3]. Our concern here is two view triangulation, which is the most basic case. As such, it has many solutions. For example, the direct linear transform method finds the point by minimizing algebraic error for the projection equations and the midpoint method finds the midpoint of the shortest segment between the backprojection rays. [4]

Maximum likelihood triangulation finds the point most likely to generate the observations with a given error model. For zero-mean isotropic Gaussian errors, this is the point for which the sum of squared projection errors to the observations is minimized. Hartley's and Sturm's [5] *optimal method* solves this problem in the two view case. The method requires finding the roots of a 6th order polynomial, but there are special cases, such as pure translational motion between the cameras, where finding the roots of a lower degree polynomial or even a linear equation suffices. One may wonder if solving a 6th degree polynomial is really necessary in the general case. Unfortunately, it is [6].

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 1–14, 2011.

Root finding for high order polynomials is a relatively expensive operation considering the ubiquity of two view triangulation. In principle, a good approximate or iterative solution can be much faster than the optimal solution while retaining optimal or near-optimal accuracy in situations of practical interest. In this paper, we consider two novel methods. The *approximate method* solves a local approximation of the optimal triangulation problem in closed form and the *local SQP method* is an iterative method based on sequential quadratic programming. Both methods perform as well as the optimal method in most cases of practical interest while being substantially faster. We also analyze the cases where the optimal method gives better results.

2 Optimal Triangulation

We start by discussing the optimal triangulation method. Our treatment here differs slightly from the original [5], but it leads to an equivalent method in fewer steps. The resulting method resembles the approximate method more closely, which helps in comparing the two. We assume that the reader is familiar with the relevant background, e.g. [5] or chapters 9 and 12 of [4], and we also try to follow the notation of the references as closely as possible.

In the optimal triangulation problem we have two cameras and two observed projections of an unknown point and the goal is to find the point whose projections minimizes the squared distances to the observations. One way to approach this problem is to try to find the point directly in space. However, the solution is easier using the *epipolar constraint*. Consider the case of two cameras and one point in space. The camera centers and the point define a plane, called an *epipolar plane*. All epipolar planes form a pencil around the axis connecting the two camera centers. The planes cut the camera images on a pair of corresponding lines, called *epipolar lines*. The projections of all points on an epipolar plane fall on these lines. Thus, only image points on corresponding epipolar lines have coplanar backprojection rays that intersect in space.

As is well known, the epipolar constraint can be expressed algebraically in terms of the image points $\mathbf{x} = (x_1, x_2, x_3)$ in the first image and $\mathbf{x}' = (x'_1, x'_2, x'_3)$ in the second image in homogeneous coordinates as

$$\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0 \tag{1}$$

where $\mathbf{F} \in \mathbb{R}^{3\times 3}$ is the *fundamental matrix* between the images. [7] From here on, we will denote analogous objects in the two views with the same symbol except with an apostrophe marking objects related to the second view. If $\mathbf{x}$ is a point in the first image, then $\mathbf{F}\mathbf{x}$ is the corresponding epipolar line in the second image, and analogously for $\mathbf{x}'$ and $\mathbf{F}^T\mathbf{x}'$. The matrix $\mathbf{F}$ is of rank 2 and its right null space corresponds to the projection of the second camera center in the first image and the left null space analogously to the the projection of the first camera center in the second image. The null spaces considered as image points are called *epipoles* and are denoted by $\mathbf{e}$ and $\mathbf{e}'$. All epipolar lines contain

the epipoles. The epipolar line corresponding to $\mathbf{x}$ in the same image is given by $[\mathbf{e}]_\times \mathbf{x}$ and analogously for the second image.

In terms of the epipolar constraint, the optimum triangulation problem is to find the closest exactly triangulable image points $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$ to the observed projections $\mathbf{x}$ and $\mathbf{x}'$, i.e.

$$\min d(\mathbf{x}, \hat{\mathbf{x}})^2 + d(\mathbf{x}', \hat{\mathbf{x}}')^2 \quad \text{subject to} \quad \hat{\mathbf{x}}'^T \mathbf{F} \hat{\mathbf{x}} = 0 \tag{2}$$

where d is the Euclidean distance. The resulting minimum is the projection error of the triangulated point. The points $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$ can be thought of as optimal corrections to $\mathbf{x}$ and $\mathbf{x}'$ so that they can be triangulated exactly. [8] We will not discuss the linear triangulation procedure to compute the final 3D point, see e.g. [5], but concentrate instead on the correction procedure before the triangulation.

The closest point on an epipolar line to an observation determines the minimum error for all points on that line. To find the nearest points, it suffices to find the pair of corresponding epipolar lines that minimizes the sum of squared distances to the observations. This can be done by finding an expression for the error in terms of an epipolar line, parameterizing the epipolar lines, and minimizing the error using this parameterization.

Suppose we have a line $\mathbf{l} = (l_1, l_2, l_3)$ and a point $\mathbf{x}$ in homogeneous form. The distance between $\mathbf{l}$ and $\mathbf{x}$ is $|\mathbf{x}^T\mathbf{l}|$ when $l_1^2 + l_2^2 = 1$ and $x_3 = 1$. For general lines and points the distance is $|\mathbf{x}^T\mathbf{l}|/\|x_3\tilde{\mathbf{I}}\mathbf{l}\|$, where $\tilde{\mathbf{I}} = \mathrm{diag}(1, 1, 0)$, and

$$\frac{\mathbf{l}^T\mathbf{x}\mathbf{x}^T\mathbf{l}}{\mathbf{l}^T(x_3^2\tilde{\mathbf{I}})\mathbf{l}} \tag{3}$$

is the squared distance. A point $\mathbf{p}$ in the first image specifies the pair of epipolar lines $[\mathbf{e}]_\times\mathbf{p}$ and $\mathbf{F}\mathbf{p}$. The error for these lines is

$$e^2(\mathbf{p}) = \frac{\mathbf{p}^T[\mathbf{e}]_\times^T\mathbf{x}\mathbf{x}^T[\mathbf{e}]_\times\mathbf{p}}{\mathbf{p}^T[\mathbf{e}]_\times^T(x_3^2\tilde{\mathbf{I}})[\mathbf{e}]_\times\mathbf{p}} + \frac{\mathbf{p}^T\mathbf{F}^T\mathbf{x}'\mathbf{x}'^T\mathbf{F}\mathbf{p}}{\mathbf{p}^T\mathbf{F}^T(x_3'^2\tilde{\mathbf{I}})\mathbf{F}\mathbf{p}} = \frac{\mathbf{p}^T\mathbf{A}\mathbf{p}}{\mathbf{p}^T\mathbf{B}\mathbf{p}} + \frac{\mathbf{p}^T\mathbf{C}\mathbf{p}}{\mathbf{p}^T\mathbf{D}\mathbf{p}} \tag{4}$$

The parameterization by a point in the first image is not that useful for optimization, because all points on an epipolar line give the same error. Choosing just one point on each epipolar line by letting $\mathbf{p} = \mathbf{x} + t\mathbf{d}$, where $\mathbf{d}$ is a direction perpendicular to the epipolar line corresponding to $\mathbf{x}$, results in

$$e^2(\mathbf{x}+t\mathbf{d}) = \frac{\mathbf{x}^T\mathbf{A}\mathbf{x} + 2t\mathbf{d}^T\mathbf{A}\mathbf{x} + t^2\mathbf{d}^T\mathbf{A}\mathbf{d}}{\mathbf{x}^T\mathbf{B}\mathbf{x} + 2t\mathbf{d}^T\mathbf{B}\mathbf{x} + t^2\mathbf{d}^T\mathbf{B}\mathbf{d}} + \frac{\mathbf{x}^T\mathbf{C}\mathbf{x} + 2t\mathbf{d}^T\mathbf{C}\mathbf{x} + t^2\mathbf{d}^T\mathbf{C}\mathbf{d}}{\mathbf{x}^T\mathbf{D}\mathbf{x} + 2t\mathbf{d}^T\mathbf{D}\mathbf{x} + t^2\mathbf{d}^T\mathbf{D}\mathbf{d}} \tag{5}$$

The minima of (5), a sum of two quadratic rational functions, can be found by differentation. Note, that due to the special choice of $\mathbf{d}$ some of the terms vanish, yielding the final form (8). The global minimum of (5) occurs at one of the real-valued zeros of a 6th degree polynomial or at $t = \infty$. The polynomial can be obtained easily from (5) using a symbolic mathematics package. The error at infinity is $e^2(\mathbf{d})$. Once the minimizing t is found, the closest points $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$ to $\mathbf{x}$

and $\mathbf{x}'$ on the corresponding epipolar lines $[\mathbf{e}]_\times(\mathbf{x}+t\mathbf{d})$ and $\mathbf{F}(\mathbf{x}+t\mathbf{d})$ (or $[\mathbf{e}]_\times\mathbf{d}$ and $\mathbf{Fd}$ for the minimum at infinity case) are the globally optimal solution to (2). Finally the pair $\hat{\mathbf{x}}, \hat{\mathbf{x}}'$ can be triangulated exactly using the linear method. In summary, we have the following procedure.

Correction Procedure for Optimal Triangulation. Given observed points $\mathbf{x}$ and $\mathbf{x}'$ in two views and the fundamental matrix $\mathbf{F}$ between the views together with the epipole $\mathbf{e}$ in the first view, find points $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$ that minimize $d(\mathbf{x}, \hat{\mathbf{x}})^2 + d(\mathbf{x}', \hat{\mathbf{x}}')^2$ subject to $\hat{\mathbf{x}}'^T\mathbf{F}\hat{\mathbf{x}} = 0$.

1. Let

$$\mathbf{A} = [\mathbf{e}]_\times^T\mathbf{x}\mathbf{x}^T[\mathbf{e}]_\times \qquad \mathbf{B} = [\mathbf{e}]_\times^T(x_3^2\tilde{\mathbf{I}})[\mathbf{e}]_\times \quad \mathbf{d} = \frac{\tilde{\mathbf{I}}[\mathbf{e}]_\times\mathbf{x}}{\|\tilde{\mathbf{I}}[\mathbf{e}]_\times\mathbf{x}\|} \tag{6}$$

$$\mathbf{C} = \mathbf{F}^T\mathbf{x}'\mathbf{x}'^T\mathbf{F} \qquad \mathbf{D} = \mathbf{F}^T(x_3'^2\tilde{\mathbf{I}})\mathbf{F} \tag{7}$$

2. Find the finite t that minimizes

$$e^2(\mathbf{x}+t\mathbf{d}) = \frac{t^2\mathbf{d}^T\mathbf{A}\mathbf{d}}{\mathbf{x}^T\mathbf{B}\mathbf{x} + t^2\mathbf{d}^T\mathbf{B}\mathbf{d}} + \frac{\mathbf{x}^T\mathbf{C}\mathbf{x} + 2t\mathbf{d}^T\mathbf{C}\mathbf{x} + t^2\mathbf{d}^T\mathbf{C}\mathbf{d}}{\mathbf{x}^T\mathbf{D}\mathbf{x} + 2t\mathbf{d}^T\mathbf{D}\mathbf{x} + t^2\mathbf{d}^T\mathbf{D}\mathbf{d}} \tag{8}$$

   by solving a 6th order polynomial and checking the values at real roots.
3. If

$$e^2(\mathbf{x}+t\mathbf{d}) < \frac{\mathbf{d}^T\mathbf{A}\mathbf{d}}{\mathbf{d}^T\mathbf{B}\mathbf{d}} + \frac{\mathbf{d}^T\mathbf{C}\mathbf{d}}{\mathbf{d}^T\mathbf{D}\mathbf{d}} \tag{9}$$

   set $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}$' to the nearest points to $\mathbf{x}$ and $\mathbf{x}'$ on the lines $[\mathbf{e}]_\times(\mathbf{x}+t\mathbf{d})$ and $\mathbf{F}(\mathbf{x}+\mathbf{d})$, respectively. Otherwise, use the lines $[\mathbf{e}]_\times\mathbf{d}$ and $\mathbf{Fd}$ instead.

3 Approximate Triangulation

The optimal solution is quite demanding from a computational point of view. One option is to simplify the problem and to solve a closely related easier problem instead. For example, most numerical iterative processes repeatedly solve locally linearized versions of nonlinear problems. Here we consider an approximate method that solves a related easier problem. First, consider the geometry of optimal triangulation. In every optimal solution, the optimal point must be the closest point to the observation on an epipolar line. The segment connecting the closest point on a line to a point is perpendicular to that line, so the closest point $\hat{\mathbf{x}}$, the observed point $\mathbf{x}$ and the epipole $\mathbf{e}$ form a right-angled triangle. By Euclid III.21, the locus of points $\hat{\mathbf{x}}$ that produce a right-angled triangle together with $\mathbf{e}$ and $\mathbf{x}$ is the circle with the segment from $\mathbf{e}$ to $\mathbf{x}$ as a diameter (Fig. 1).

The circle containing the optimal solution can be approximated by its tangent through $\mathbf{x}$ in the neighborhood of $\mathbf{x}$. The distance along the tangent is a good approximation to the actual distance when $\mathbf{e}$ is relatively farther from $\mathbf{x}$ than $\hat{\mathbf{x}}$ is. The approximate method minimizes the squared distances along the tangents. Let $\mathbf{d}$ and $\mathbf{d}'$ be unit length direction vectors of the circles' tangents through $\mathbf{x}$ and $\mathbf{x}'$.

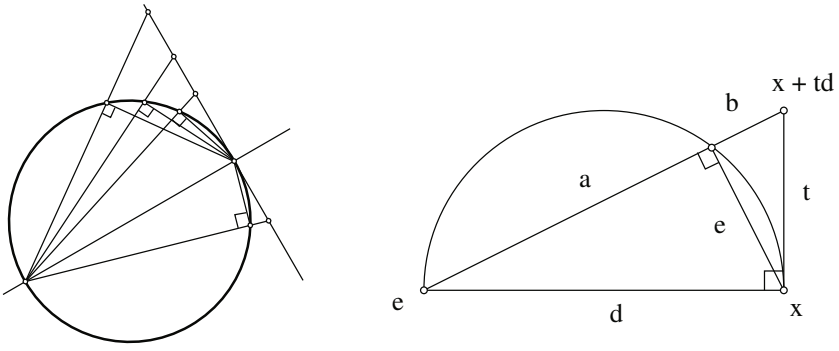


Fig. 1. Left: Locus of solutions to the optimal triangulation problem in one image. In the other image the situation is analogous. All solutions lie on a circle connecting the observation and the epipole. Right: Geometric relationship between the true error e and the approximate error t minimized in the approximate method.

Normalize the points so that $x_3 = x'_3 = 1$. Now, $\mathbf{x} + t\mathbf{d}$ is t units away from $\mathbf{x}$, $\mathbf{x}' + t'\mathbf{d}'$ is t' units away from $\mathbf{x}'$, and the epipolar constraint is satisfied, iff

$$(\mathbf{x}' + t'\mathbf{d}')^T\mathbf{F}(\mathbf{x} + t\mathbf{d}) = 0 \iff t' = -\frac{\mathbf{x}'^T\mathbf{F}\mathbf{x} + t\mathbf{x}'^T\mathbf{F}\mathbf{d}}{\mathbf{d}'^T\mathbf{F}\mathbf{x} + t\mathbf{d}'^T\mathbf{F}\mathbf{d}} \tag{10}$$

For given values of t and t', the approximate error is $t^2 + t'^2$. The minimum approximate error for points on the tangents is the global minimum of

$$t^2 + t'^2 = t^2 + \left(\frac{\mathbf{x}'^T\mathbf{F}\mathbf{x} + t\mathbf{x}'^T\mathbf{F}\mathbf{d}}{\mathbf{d}'^T\mathbf{F}\mathbf{x} + t\mathbf{d}'^T\mathbf{F}\mathbf{d}}\right)^2 \tag{11}$$

which can be found at one of the real roots of a 4th degree polynomial. These roots have closed form expressions and can be found with the approximately the same amount of computation in all cases. Picking the nearest points to $\mathbf{x}$ and $\mathbf{x}'$ on the epipolar lines corresponding to the minimum further decreases the error. This gives the following procedure.

Approximate Correction Procedure for Triangulation. Given observed points $\mathbf{x}$ and $\mathbf{x}'$ in two views and the fundamental matrix $\mathbf{F}$ between the views together with the epipoles $\mathbf{e}$ and $\mathbf{e}'$, find points $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$ that approximately minimize $d(\mathbf{x}, \hat{\mathbf{x}})^2 + d(\mathbf{x}', \hat{\mathbf{x}}')^2$ subject to $\hat{\mathbf{x}}'^T\mathbf{F}\hat{\mathbf{x}} = 0$.

1. Normalize $\mathbf{x}$ and $\mathbf{x}'$ so that $x_3 = x'_3 = 1$ and let

$$\mathbf{d} = \frac{\tilde{\mathbf{I}}[\mathbf{e}]_\times\mathbf{x}}{\|\tilde{\mathbf{I}}[\mathbf{e}]_\times\mathbf{x}\|} \qquad \mathbf{d}' = \frac{\tilde{\mathbf{I}}[\mathbf{e}']_\times\mathbf{x}'}{\|\tilde{\mathbf{I}}[\mathbf{e}']_\times\mathbf{x}'\|}$$

2. Find t that minimizes

$$t^2 + \left(\frac{\mathbf{x}'^T \mathbf{F} \mathbf{x} + t \mathbf{x}'^T \mathbf{F} \mathbf{d}}{\mathbf{d}'^T \mathbf{F} \mathbf{x} + t \mathbf{d}'^T \mathbf{F} \mathbf{d}} \right)^2$$

The minimum occurs at one of the real roots of

$$d^3 t^4 + 3cd^2 t^3 + 3c^2 dt^2 + (c^3 + b^2 c - abd)t + abc - a^2 d$$

where $a = \mathbf{x}'^T \mathbf{F} \mathbf{x}$, $b = \mathbf{x}'^T \mathbf{F} \mathbf{d}$, $c = \mathbf{d}'^T \mathbf{F} \mathbf{x}$, and $d = \mathbf{d}'^T \mathbf{F} \mathbf{d}$.

3. Set $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}$' to the nearest points to $\mathbf{x}$ and $\mathbf{x}'$ on the lines $[\mathbf{e}]_\times(\mathbf{x} + t\mathbf{d})$ and $\mathbf{F}(\mathbf{x} + t\mathbf{d})$, respectively.

3.1 Theoretical Performance Bounds

For an approximate method to be useful it must have some guarantee on the quality of approximation. First, consider the method in the limit as the approximation gets closer to the actual error. Denote by d the distance from $\mathbf{x}$ to $\mathbf{e}$ and by e the distance from $\mathbf{x}$ to the epipolar line through $\mathbf{x} + t\mathbf{d}$. Connect $\mathbf{x} + t\mathbf{d}$ by a segment to $\mathbf{e}$. Divide the segment into two parts by a perpendicular line to $\mathbf{x}$ and let a and b be the lengths of the parts (Fig. 1). By equality of areas and by the Pythagorean theorem $e(a + b) = dt$ and $d^2 + t^2 = (a + b)^2$. We have

$$e^2 = \frac{t^2}{1 + (t/d)^2} \qquad e'^2 = \frac{t'^2}{1 + (t'/d')^2} \tag{12}$$

Now $e^2 \to t^2$ as $d \to \infty$. The optimal method minimizes

$$e^2 + e'^2 = \frac{t^2}{1 + (t/d)^2} + \frac{t'^2}{1 + (t'/d')^2} \to t^2 + t'^2 \tag{13}$$

as $\min(d, d') \to \infty$. The approximate method converges to the optimal method when both epipoles move to infinity. Epipoles at infinity is a typical case that occurs with parallel, non-convergent cameras, for example.

Next, we consider approximation bounds. Denote by $\tilde{t}$ and $\tilde{t}'$ the case with minimum approximate error, by $\tilde{e}$ and $\tilde{e}'$ the resulting final error, and by $\hat{e}$, $\hat{e}'$ the optimal error. The approximate error corresponding to the optimal solution bounds the minimum approximate error from above, so that

$$\tilde{t}^2 + \tilde{t}'^2 \leq \frac{\hat{e}^2}{1 - (\hat{e}/d)^2} + \frac{\hat{e}'^2}{1 - (\hat{e}'/d')^2} \leq \frac{\hat{e}^2 + \hat{e}'^2}{1 - (\hat{e}^2 + \hat{e}'^2)/\min(d, d')^2} = B \tag{14}$$

where the latter bound is valid, when $\hat{e}^2 + \hat{e}'^2 \leq \min(d, d')^2$. The approximate error quickly converges to the optimal error, when the maximum error in the images decreases or the minimum distance to an epipole increases.

Picking closest points on the epipolar lines also decreases the error and makes the method more accurate. The maximum error attainable given $\hat{e}^2 + \hat{e}'^2$ bounds the final error from above for any particular case. Thus,

$$\tilde{e}^2 + \tilde{e}'^2 \leq \max_{t^2+t'^2 \leq B} \frac{t^2}{1+(t/d)^2} + \frac{t'^2}{1+(t'/d')^2} = \frac{B}{1+B/(d^2+d'^2)} \tag{15}$$

$$= \frac{\hat{e}^2 + \hat{e}'^2}{1 + \left(\dfrac{1}{d^2+d'^2} - \dfrac{1}{\min(d,d')^2}\right)(\hat{e}^2 + \hat{e}'^2)} \tag{16}$$

The maximization in (15) requires a bit of calculus. The effect of picking closest points on epipolar lines is most prominent in symmetric cases, where the epipoles are outside the images at approximately the same distance. In these cases it increases the effect of increasing the distance to epipoles by a factor of $\sqrt{2}$. As an example, a pair of slightly convergent cameras is such a symmetric case.

In summary, the approximate method is close to optimal when the optimal error is small compared to the distances from the observed points to the epipoles and performs best, when the observations are almost equidistant from epipoles. The worst case bound is (16). Worst cases occur when the error is concentrated in one image in the optimal solution. Intuitively this is not very likely.

4 Local SQP Correction

The two previous methods were tailored specifically for the optimal triangulation problem, but it can also be tackled fairly easily using standard numerical tools. We consider a local sequential quadratic programming (SQP) solution to (2). In short, we solve

$$\begin{bmatrix} \nabla(d(\hat{\mathbf{x}},\mathbf{x})^2 + d(\hat{\mathbf{x}}',\mathbf{x}')^2 - \lambda \hat{\mathbf{x}}'^T \mathbf{F} \hat{\mathbf{x}}) \\ \hat{\mathbf{x}}'^T \mathbf{F} \hat{\mathbf{x}} \end{bmatrix} = 0 \tag{17}$$

for $\hat{\mathbf{x}}$, $\hat{\mathbf{x}}'$, and λ using Newton's method. The gradient is taken with respect to $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$, of course. The SQP method converges quickly when the iterations are started near a solution or when the problem is otherwise sufficiently simple. For more details on SQP, see e.g. Chapter 18 of [9]. In practical cases of two view triangulation the optimal triangulation error should be small, so the original observed points provide a good starting point for the iterations. Normalizing input points so that $x_3 = x'_3 = 1$ and optimizing over the first two coordinates gives the method below.

SQP Correction Procedure for Triangulation. Given observed points $\mathbf{x}$ and $\mathbf{x}'$ (with $x_3 = x'_3 = 1$) in two views and the fundamental matrix $\mathbf{F}$ between the views, find points $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$ that minimize $d(\mathbf{x},\hat{\mathbf{x}})^2 + d(\mathbf{x}',\hat{\mathbf{x}}')^2$ subject to $\hat{\mathbf{x}}'^T \mathbf{F} \hat{\mathbf{x}} = 0$.

1. Let $\hat{\mathbf{x}}_1 = \mathbf{x}$, $\hat{\mathbf{x}}'_1 = \mathbf{x}'$, $\lambda_1 = 0$.
2. Repeat for $k = 1, 2, \ldots$: Solve the SQP step from

$$\begin{bmatrix} 2\mathbf{I} & -\lambda_k \mathbf{F}^T & -\mathbf{F}^T\hat{\mathbf{x}}'_k \\ -\lambda_k \mathbf{F} & 2\mathbf{I} & -\mathbf{F}\hat{\mathbf{x}}_k \\ \hat{\mathbf{x}}'^T_k \mathbf{F} & \hat{\mathbf{x}}^T_k \mathbf{F}^T & 0 \end{bmatrix} \begin{bmatrix} \Delta\hat{\mathbf{x}}_k \\ \Delta\hat{\mathbf{x}}'_k \\ \lambda_{k+1} \end{bmatrix} = - \begin{bmatrix} 2(\hat{\mathbf{x}}_k - \mathbf{x}) \\ 2(\hat{\mathbf{x}}'_k - \mathbf{x}') \\ \hat{\mathbf{x}}'^T_k \mathbf{F}\hat{\mathbf{x}}_k \end{bmatrix} \tag{18}$$

   where all non-scalar component expressions are finally truncated to 2×2 matrices or 2-vectors (to remove the constant homogeneous part). Set

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_k + \Delta\hat{\mathbf{x}}_k \qquad \hat{\mathbf{x}}'_{k+1} = \hat{\mathbf{x}}'_k + \Delta\hat{\mathbf{x}}'_k \tag{19}$$

   Stop, when $d(\hat{\mathbf{x}}_{k+1}, \hat{\mathbf{x}}_k)^2 + d(\hat{\mathbf{x}}'_{k+1}, \hat{\mathbf{x}}'_k)^2 < \epsilon$.
3. The corrected points are $\hat{\mathbf{x}}_{k+1}$ and $\hat{\mathbf{x}}'_{k+1}$.

The SQP method requires solving a 5×5 linear system. For a practical implementation, the solution of (18) can be obtained symbolically with a block LDU-decomposition of the matrix. Multiplication by the inverse of this decomposition is easy to turn into code, as the decomposition leaves one non-trivial 2×2 matrix and 1 scalar on the diagonal and the rest of the factors are very easy to invert. A practical implementation should also have safeguards against singular systems for the step, e.g. at the epipoles, and slow convergence, though both of these occur extremely rarely in practice. In empirical tests, we used $\epsilon = 10^{-6}$ for the stopping criterion.

5 Empirical Performance Evaluation

We ran simulations comparing the optimal, approximate and SQP methods on the exact same data to determine their performance differences in practice. It is of course impossible to cover all possible cases in simulation, so we focused on three types of situations with varying parameters. The first is turntable motion (A), where the cameras converge on an object from a circle surrounding it, the second is translation (B), where the camera centers are not coplanar, and the third is an asymmetric situation (C), where the camera centers are coplanar and the second camera is rotated toward the first camera. In situation (C) one of the epipoles is at infinity. Finally we tested the methods on real world data to validate the simulation results.

5.1 Simulation Setup

We implemented the methods in MATLAB and in C/C++ using the MEX interface and tested their performance on simulated data. In test set (A) two normalized cameras observe the unit ball from a circle of radius 2. The cameras converge on the origin and there are five different positions of the second camera. The camera centers are situated on the circle 11.25° (A1), 22.5° (A2), 45° (A3), 90° (A4), and 180° (A5) degrees apart. In set (B), the second camera is two units

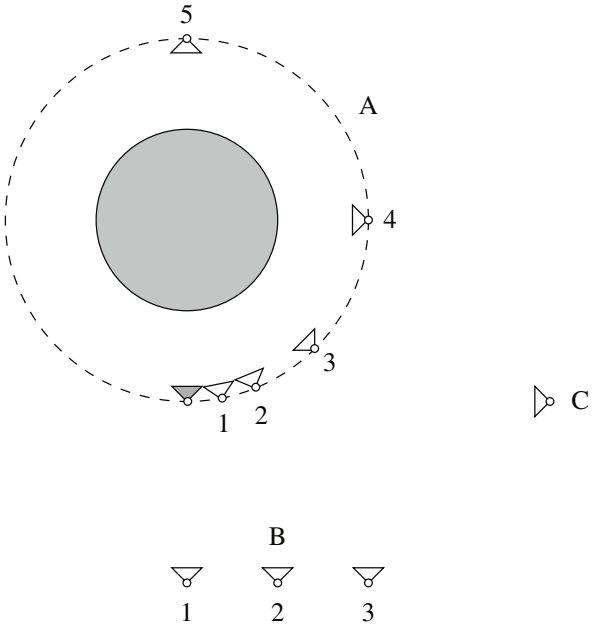


Fig. 2. The test setup for empirical performance evaluation as seen from above. The cameras observe points picked at random in the unit ball with a random observation error. The reference camera (filled) stays in the same place and the position of the second camera varies. The test sets consist of turntable motion (A), pure translation (non-coplanar) (B), and an asymmetric case (C).

behind the first camera and displaced by 0 (B1), 1 (B2), and 2 (B3) units to the side. In test set (C), the second camera is 4 units to the side and pointed at the first camera. Fig. 2 depicts the situation from above. In addition, we tested cases where the camera centers are coplanar and the principal directions are parallel to check that the methods are identical in the case when epipoles are at infinity. This proved to be the case, so these results are not reported.

For the observations, we generated $N = 100000$ points at random from an uniform distribution on the unit ball using rejection sampling: Pick a point at uniform random over the cube $[-1, 1]^3$ and discard it if it is outside the unit ball. The cameras then observe the 3D points with zero-mean isotropic Gaussian noise of deviation σ from 0 to 0.2 added to the projections. The range is quite extreme as in typical application cases $\sigma \leq 0.01$, but the results will show where the approximate method breaks. Theoretically the total squared displacement in the test cases is a χ^2 random variable with 4 degrees of freedom scaled by σ^2. The total mean squared displacement of an observation pair is $4\sigma^2$ and in a single image the mean squared displacement is $2\sigma^2$.

In the tests, we then used the optimal, approximate and SQP algorithms to correct the noisy observations using the exact same data and observed the *triangulation error*, the sum squared distances to the projections, and the *estimation error*, the sum of squared distances to the error-free projections.

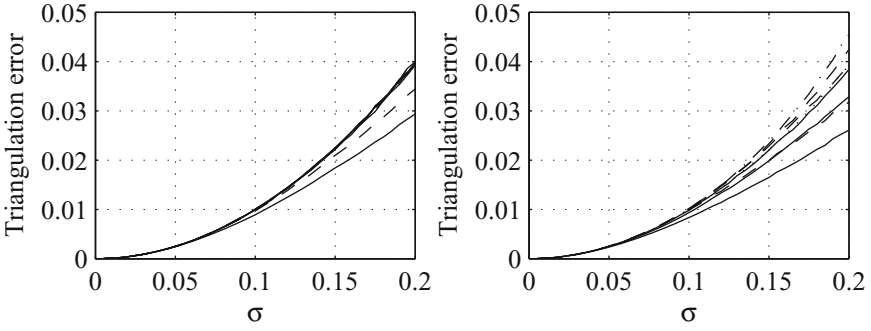


Fig. 3. Triangulation error vs. noise level. Cases A1-A5 are shown on the left with the approximate method dashed. Case A5 has the smallest error. Cases B1-B3 and C are shown on the right. Case B1 has the smallest error. In case C, the optimal result is dotted and approximate result dash-dotted.

5.2 Simulation Results

The mean triangulation and estimation errors of the methods were very close in cases with small observation errors. The local SQP method's average triangulation errors were ≤ 1.0002 times the corresponding optimal method's triangulation error in all cases, so there is little point in presenting the results separately. Fig. 3 contains the mean triangulation errors of the optimal and approximate methods for the test sets. The final triangulation error of the optimal method is about $1/4$ of the theoretical mean displacement, and the mean estimation error of the optimal method was about $3/4$ of the theoretical mean displacement. There is a much smaller difference in estimation errors than in the triangulation errors (Fig. 4). The approximate method is more accurate than the optimal method in cases A1-A5, worse in cases B1-B3, and worst in case C.

Fig. 5 depicts ratios of mean triangulation errors of the approximate and the optimal method for the test sets. When observation errors are small, $\sigma \in [0, 0.005]$, the approximate method's errors are ≤ 1.001 times optimal, with cases $A5$ and $B1$ clearly the worst. These cases have epipoles right in the middle of the image points. In the other cases the errors were much closer to optimal. The ratios increase differently depending on the situation as the observation error increases. In the case of estimation errors the ratios fluctuate from 0.98 to 1.03.

5.3 Computational Efficiency

Computational efficiency was one of the motivations for the methods, so we compared the efficiency of the methods on the test sets. A rigorous comparison of the methods is difficult, because, for example, the efficiency of the optimal method depends on the polynomial root finder and its efficiency on the particular test cases. All methods were implemented in C++ with about an equal amount of manual tuning. The abstract descriptions do not give the most efficient ways to implement the methods, e.g. the ordering of matrix multiplications matters and some of the multiplications are cross and dot products.

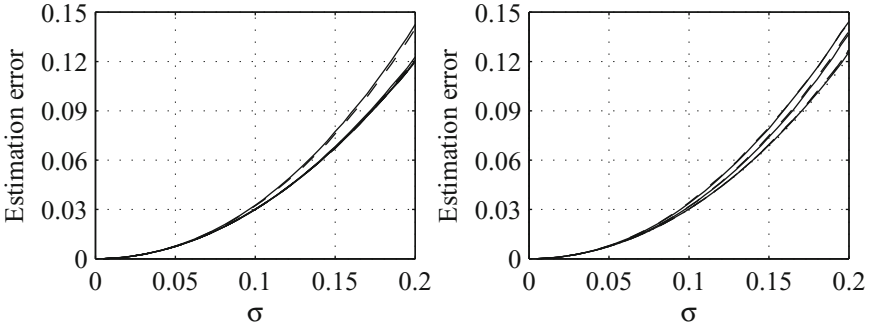


Fig. 4. Estimation error vs. noise level. Cases A1-A5 are shown on the left with the approximate method dashed. Case A1 has the smallest error. Cases B1-B3 and C are shown on the right. Case B3 has the smallest error. In case C, the optimal result is dotted and approximate result dash-dotted.

We tested two different polynomial root finders for the optimal method. The first method finds the eigenvalues of the polynomial's companion matrix. Our implementation used LAPACK's `DGEES` subroutine called directly from C++. The second polynomial solver used was `gsl_poly_complex_solve` from the GNU Scientific Library (GSL). For the approximate method, we used the closed form Ferrari's method to find the roots of the quartic. The optimal and approximate methods required a test for the order of the polynomial to be solved. In special cases the higher order polynomial coefficients that should have been zero were very small due to roundoff error, which caused havoc in the polynomial solver.

We measured the clock cycles taken by each method on the test sets using the `rdtsc` instruction on a Intel Core 2 Quad processor. The SQP method was the fastest with its speed inversely related to the amount of noise added to the observations; larger optimal errors required more SQP iterations until convergence. The camera configuration also had an effect. In cases (A) and (B) the approximate method was 2.8-5 times slower than SQP and the optimal method with GSL was about 100-150 times slower and with DGEES about 200-400 times slower. In test set (C) the optimal method with GSL was 50 times slower than SQP; the optimal method needed to find the roots of a simpler 5th degree polynomial instead of a 6th degree polynomial. In the special case of parallel cameras the polynomial in the optimal solution is of 1st degree, and the optimal method is only about 1.5 times slower than SQP.

5.4 Validation with Real World Data

Simulation results should be validated with real world data. To obtain suitable data, we reconstructed the Leuven castle image sequence [10] from SIFT[11] features and recorded all two view triangulations that occurred during the reconstruction process. The resulting reconstruction shown in Fig. 6 consists of 28 cameras, 8651 points, and 66611 projections giving 7.7 projections per

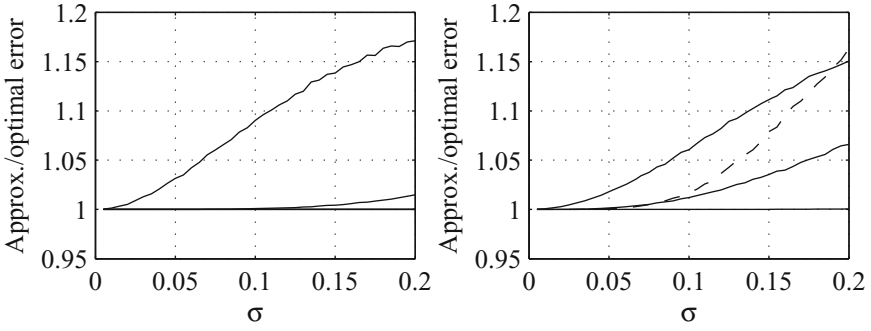


Fig. 5. Ratios of mean triangulation error of the approximate and the optimal error vs. noise level. Cases A1-A5 on the left in increasing order (cases A1-A3 indistinguishable) and cases B1-B3 in decreasing order on the right with case C dashed.

point and 2379 projections per camera on the average. In the final result, the median projection error in the images is 0.17 pixels. Our reconstruction pipeline performed 73091 two view triangulations during the reconstruction process.

We repeated the recorded triangulations using the optimal, approximate, and SQP algorithms and observed the triangulation errors. The optimal and SQP methods produced nearly identical results: All SQP triangulation errors were smaller than 1.00003 times optimal. SQP took typically 2–3 iterations and at most 6 iterations to converge. Almost all, 99.96%, of the approximate method's

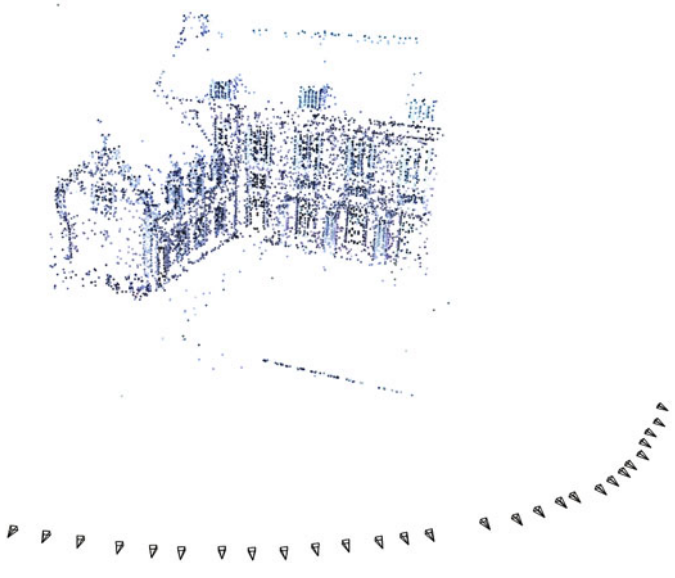


Fig. 6. A sparse metric reconstruction of the Leuven castle data set from SIFT features. Two view triangulations that occurred during the reconstruction process were used to evaluate the triangulation methods.

triangulation errors were smaller than 1.0001 times optimal, and only 4 cases were larger than 1.01 times optimal with a maximum of 1.9 times optimal.

In the reconstruction pipeline we used the correction procedure to check if a feature match is an inlier for a given **F** while estimating **F** for the first image pair using RANSAC. The difficult cases had large optimal triangulation errors and originated from this stage of the process: Either **F** was wrong or the match was incorrect. A different inlier criterion, for example the matching points' distances to each other's epipolar lines, could be used to avoid these cases. The results on this data set agree with the simulation results.

6 Conclusions and Discussion

We presented two novel methods for two view triangulation. The approximate method minimizes a local approximation to the optimal triangulation problem. The SQP method solves the optimal triangulation problem iteratively using sequential quadratic programming. Finally, we tested the triangulation methods using simulations and real world data obtained from a reconstruction process.

The test results for the approximate method are in line with the theoretical bounds: The triangulation errors are very close to optimal in cases with small observation error. The cases with the largest differences correspond to situations where the epipoles are near the object's images and the observation error is large. For most cases of practical interest the methods give almost identical results. For example, in reconstruction from SIFT [11] or SURF [12] feature points extracted from photographs taken with the same camera, the final error of an observation is usually on the order of one pixel. This corresponds to errors with approximately $\sigma = 0.0005$ for 3072×2304 images, for example, given that the mean residual error is somewhat smaller than the actual error.

While the approximate method produces slightly larger triangulation errors than the optimal method, it was in cases slightly more accurate in the sense of estimating the true projections. The reason may be bias. Considering the locus of solutions in Fig. 1 it seems probable that the optimal method produces corrections that are very slightly biased toward the epipoles, because when epipoles are finite every non-zero correction is closer to the epipoles than the original observations. If this geometric reasoning is valid, then the method should be unbiased when the epipoles are at infinity and the bias should decrease when the observation error decreases and the distance to epipoles increases. A plausible bias correction would be to pick closest points on some other curves, perhaps on circles centered on the epipoles containing the observations. The benefit is likely to be negligible in most practical cases.

The local SQP method seems to be the fastest and produces results that are as good as the optimal method, at least on the average. In some rare cases the method finds a suboptimal solution. These occur when the epipoles are inside the images and the observation errors can substantially change the projections relative to the epipoles, e.g. to the other side of the epipole. In these cases the optimal method will find the global optimum, but the optimum is still wildly

inaccurate as an estimate. The method in [8] can be obtained by ignoring the cross terms $-\lambda_k \mathbf{F}^T$ and $-\lambda_k \mathbf{F}$ in the Hessian of the Lagrangian in (18), which simplifies the iterations, but changes the convergence properties; a theoretical comparison of the two methods would be interesting.

In summary, the approximate method is a reasonable alternative for the optimal method, especially if the observation errors are known to be small and constant computation time is desired. It is a closed form approximation to the optimal method that works well when the optimal solution is near the observations and the observations are far from the epipoles. In terms of estimation accuracy, the optimal and approximate methods are about equal. It seems that the optimal method is most useful near the epipoles when the optimal error is relatively large, but the triangulation results in these cases are of limited value in practical applications. Given the simulation results, the local SQP method is in our opinion the best choice of the three for most applications. It should also generalize fairly easily to more than two views.

Acknowledgement. This work was partially funded by Tivit (DIEM/MMR project) and by the TKK MIDE programme (UI-ART project).

References

1. Kahl, F., Hartley, R.: Multiple-view geometry under the L_∞-norm. IEEE Transactions on Pattern Analysis and Machine Intelligence 30, 1603–1617 (2008)
2. Byröd, M., Josephson, K., Åström, K.: Fast optimal three view triangulation. In: Yagi, Y., Kang, S.B., Kweon, I.S., Zha, H. (eds.) ACCV 2007, Part II. LNCS, vol. 4844, pp. 549–559. Springer, Heidelberg (2007)
3. Nordberg, K.: The Triangulation Tensor. Computer Vision and Image Understanding 113, 935–945 (2009)
4. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2003)
5. Hartley, R.I., Sturm, P.: Triangulation. Computer Vision and Image Understanding 68, 146–157 (1997)
6. Nister, D., Hartley, R., Stewenius, H.: Using Galois Theory to Prove Structure from Motion Algorithms are Optimal. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2007). IEEE Computer Society, Los Alamitos (2007)
7. Luong, Q.T., Faugeras, O.: The Fundamental matrix: theory, algorithms, and stability Analysis. International Journal of Computer Vision 17, 43–75 (1995)
8. Kanatani, K., Sugaya, Y., Niitsuma, H.: Triangulation from two views revisited: Hartley-Sturm vs. optimal correction. In: Proceedings of the 19th British Machine Vision Conference (BMVC 2008), pp. 173–182 (2008)
9. Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edn. Springer, Heidelberg (2006)
10. Pollefeys, M., Van Gool, L., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., Koch, R.: Visual modeling with a hand-held camera. International Journal of Computer Vision 59, 207–232 (2004)
11. Lowe, D.G.: Object recognition from local scale-invariant features. In: ICCV 1999: Proceedings of the International Conference on Computer Vision, Washington, DC, USA, pp. 1150–1157. IEEE Computer Society, Los Alamitos (1999)
12. Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: SURF: Speeded Up Robust Features. Computer Vision and Image Understanding 110, 346–359 (2008)

Adaptive Motion Segmentation Algorithm Based on the Principal Angles Configuration

L. Zappella[1], E. Provenzi[2], X. Lladó[1], and J. Salvi[1]

[1]Institut d'Informàtica i Aplicacions, Universitat de Girona, Girona, Spain
[2]Departamento de Tecnologías de la Información y las Comunicaciones, Universitat Pompeu Fabra, Barcelona, Spain

Abstract. Many motion segmentation algorithms based on manifold clustering rely on a accurate rank estimation of the trajectory matrix and on a meaningful affinity measure between the estimated manifolds. While it is known that rank estimation is a difficult task, we also point out the problems that can be induced by an affinity measure that neglects the distribution of the principal angles. In this paper we suggest a new interpretation of the rank of the trajectory matrix and a new affinity measure. The rank estimation is performed by analysing which rank leads to a configuration where small and large angles are best separated. The affinity measure is a new function automatically parametrized so that it is able to adapt to the actual configuration of the principal angles. Our technique has one of lowest misclassification rates on the Hopkins155 database and has good performances also on synthetic sequences with up to 5 motions and variable noise level.

1 Introduction

Given a cloud of features tracked throughout a video sequence, the motion segmentation problem consists of clustering together features that follow the same movement. Such a problem is a fundamental step for many computer vision tasks like robotics, inspection, video surveillance, and many other applications. Motion segmentation has become even more important after the introduction of the structure from motion algorithms, which can mostly deal with only one motion at a time [8].

A 3D cloud of P points that belong to N independent and rigid motions can be mapped onto the video sequence through affine projection. The 2D position of each point at each frame can be stored into a *trajectory matrix* $\mathbf{W} \in \mathbb{R}^{2F \times P}$, where F is the total number of frames of the input sequence. Assuming no noise and no outliers, most of the rigid motion segmentation algorithms based on manifold clustering rely on a simple assumption: each independent motion generates a *local subspace* of size at most 4, therefore the union of the local subspaces generates a *global subspace* of size at most $4N$, size that corresponds to the rank of $\mathbf{W}$.

Two main ideas distinguish the majority of motion segmentation algorithms based on manifold clustering that can be found in the literature: the first is

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 15–26, 2011.

how they estimate the manifold generated by each trajectory, the second is how they group them through the selection of *suitable common properties*.

Related Works on Motion Segmentation via Manifold Clustering. In [10] the authors use the Generalized Principal Component Analysis (GPCA) in order to fit a polynomial of degree N to the data, where N is the number of subspaces. Then, they estimate the basis of the subspaces using the derivatives of the polynomial and they build a similarity matrix based on the $\cos^2$ function of the principal angles (PAs) between the subspaces. Another way for the subspace estimation is via the singular value decomposition (SVD) of $\mathbf{W}$, like in the Local Subspace Affinity [11] framework (LSA). LSA also uses the PAs between subspaces in order to build the affinity matrix, however, LSA adopts a different similarity function. An Enhanced LSA (ELSA) is proposed in [12] where one of the improvements is a more robust model selection for the estimation of the global subspace size. Also in [6] the dimension of the global subspace is at the center of the study, they suggest lower and upper bounds together with a data-driven procedure for choosing the optimal ambient dimension. In [3], a new way for describing the subspaces called Sparse Subspace Clustering (SSC) is presented. The authors exploit the fact that each point (in the global subspace) can be described with a sparse representation (obtained by an ℓ_1 optimization) with respect to the dictionary composed by all of the points. The final similarity matrix is built using the coefficients of the sparse representation. Another idea is used in [4], where the authors propose a subspace segmentation algorithm based on a Grassmannian minimization approach. The estimation of the subspaces is performed via the Maximum Consensus Subspace (MCS) criteria. The same framework is further extended by using the Normalized Subspace Inclusion (NSI) similarity measure [5] between the PAs of the estimated subspaces. The Agglomerative Lossy Compression (ALC) algorithm [7] differs from the previous methods in that it does not require a similarity matrix. ALC is an agglomerative strategy that consists of minimizing the segmentation coding length in order to find the shortest coding length which is theoretically the optimal.

All of these techniques rely on the ability of the algorithms to estimate the subspaces and then to compare them (with exception of ALC). As shown in [9, 12] the size estimation of the global subspace (when required) is a critical and very difficult step. Moreover, the similarity measures used until now are *rigid* as they always assume that the features between similar and different subspaces are well separated. However, we show in section 2.4 that such an assumption is not always verified.

Our Contribution. In this work we provide two main contributions: a new interpretation of the global subspace size estimation and a new similarity measure between subspaces. Our new subspace size estimation does not depend on any sensitive parameter, and it is able to select the dimension of the global subspace where the distribution of the PAs is the most suited for the clustering step. Moreover, our similarity measure is able to dynamically adapt to the distribution of the PAs.

The results of these two contributions are evaluated on the LSA framework. We compared our model with some state of the art techniques [3, 5, 7, 9, 11, 12] on the Hopkins155 database [9], showing that our proposal outperforms all of the LSA-based algorithms, providing one of the lowest misclassification rate in the literature. Our method will be also applied on synthetic sequences from 2 to 5 motions with a controlled noise level in order to test the robustness against noise and the behaviour with more than 3 motions. Matlab source code of our algorithm can be found at: `http://eia.udg.es/~zappella`.

2 Our Proposal

In this section we present a new rank estimation for $\mathbf{W}$ based on the clusterization level of the principal angles and a new adaptive similarity measure for principal angles. We apply these two techniques to the LSA framework as it is theoretically able to deal with different types of motion: independent, articulated, rigid, non-rigid, degenerate and non-degenerate. Before going into the detail of our proposal we introduce a convenient notation and we discuss some issues regarding the principal angles.

2.1 Notation

Given a collection of N subspaces, the PAs between two subspaces S_j and S_l, for $j, l = 1, \ldots, N$, are defined recursively as a series of angles $0 \leq \theta_1 \leq \ldots \leq \theta_i \leq \ldots \leq \theta_M \leq \pi/2$, where $M = \min\{rank(S_j), rank(S_l)\}$:

$$\begin{aligned} \cos(\theta_1) &= \max_{u \in S_j, v \in S_l} u^T v = u_1^T v_1 \\ \cos(\theta_i) &= \max_{u \in S_j, v \in S_l} u^T v = u_i^T v_i, \forall i = 2, \ldots M \end{aligned} \tag{1}$$

such that: $\|u\| = \|v\| = 1$, $u^T u_j = 0$, $v^T v_j = 0$, $\forall j = 1, \ldots, i-1$. The vectors $u_1, \ldots, u_i$ and $v_1, \ldots, v_i$ are the principal vectors (u and v being two generic principal vectors). We denote with:

$$\theta_i^r(S_j, S_l) \tag{2}$$

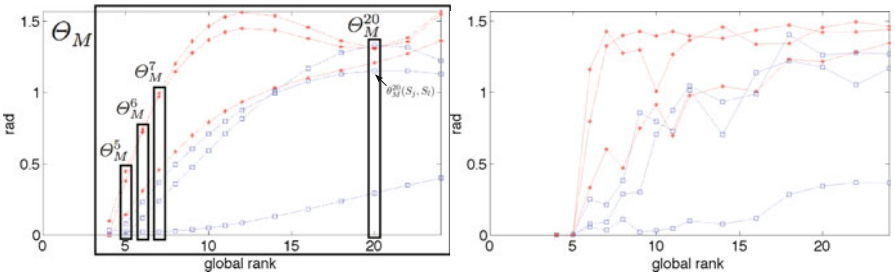
the i^{th} PA between the subspaces S_j and S_l computed when the estimated size of the global subspace is r. As j and l vary we define the set:

$$\Theta_i^r = \{\theta_i^r(S_j, S_l),\ j, l = 1, \ldots, P\} \tag{3}$$

Finally, we define:

$$\Theta_i = \bigcup_{r=1}^{r_{\max}} \Theta_i^r \tag{4}$$

where $r_{\max}$ is the upper bound of the global subspace size. For an at-a-glance overview of our notation refer to Fig. 1a.



(a) Interpolated PAs (b) Original PAs

Fig. 1. Small random subset of the PAs of Θ_M (largest PAs) of the sequence 1R2RCT_A taken from the Hopkins155 database. PA between similar subspaces are represented with blue squares, PAs between different subspaces are represented with red asterisks.

2.2 Issues Regarding the Behaviour of Principal Angles

PAs between two subspaces are an efficient measure of orthogonality when the exact subspace bases are known. However, when the bases are estimated there are some issues that should be taken into account, especially when the exact size of the global subspace is unknown. In [12] the behaviour of PAs, computed following the LSA algorihtm, when the estimated rank r of the global subspace changes is studied. The authors explain that the trend of PAs, going from an underestimation to an overestimation of r, is overall increasing typically starting from 0 radians and ending in $\pi/2$ radians, as in Fig. 1. In the same study it is explained that despite the overall increasing trend, the PAs may have oscillations, as in Fig. 1b, due to the fact that when the rank is underestimated the bases are not well defined, while when the rank is overestimated the extra components introduced act like noise.

In order to reduce the influence of these oscillations we propose a *polynomial interpolation* of the PAs across the different ranks. We avoid the trivially useless interpolation of order 1. The interpolation of order 2 is decreasing after its maximum, this does not fit with the increasing behaviour of the PAs. The interpolation of order 3 is able to smoothly follow the PAs trend, as shown in Fig. 1a. Interpolation of higher degrees would adhere too much to the data making the interpolation not effective. We conducted different tests on synthetic and real sequences that confirm the PAs behavior and the reliability of the interpolation of order 3.

2.3 Rank Selection via Principal Angles Clusterization (PAC)

One of the most recognized weaknesses of LSA is the lack of robustness of the Model Selection (MS) procedure for the estimation of the rank r:

$$r = \underset{r}{\operatorname{argmin}} \left(\frac{\lambda_{r+1}^2}{\sum_{i=1}^{r} \lambda_i^2} + kr \right) \tag{5}$$

λ_i being the i^{th} singular value of $\mathbf{W}$, and k a parameter that depends on the noise of the tracked point positions. Eq. (5), is extremely sensitive to changes of the parameter k. On the other hand, k is necessary in order to deal with sequences with different amounts of noise and number of motions. In [9] the authors decided to avoid the use of MS due to the difficult task of finding a value of k that could cope with all of the sequences of the Hopkins155 database. Therefore, they fixed the global subspace size to $4N$. Fixing the global subspace size to $4N$ implies that the motions are all rigid and fully independent. Such an assumption reduces the efficiency of LSA. In order to solve this problem in [12] the authors present an algorithm named ELSA with an Enhanced Model Selection (EMS+). EMS+ consists of computing different affinity matrices, by using different k values with the MS formula, and selecting the affinity matrix with the maximum entropy. This technique allows homogeneous affinity matrices (which correspond to over- or underestimation of the rank) to be discarded, and to use an affinity matrix with the highest content of information. ELSA with EMS+ performs better than LSA with MS. Nevertheless, as the authors explain, EMS+ tends to underestimate the rank and it fails in the ideal case when the affinity matrix is binary.

The problem of the rank estimation in real cases, with noise and dependent motions, is challenging because the eigenvalue spectrum of $\mathbf{W}$ tends to become smooth and the selection of a threshold becomes a difficult task. Therefore, we decided to renounce the computation of the rank in the traditional way and we studied the distribution of the PAs in each Θ_i. The fundamental idea on which our proposal is based is that *the rank r should be selected, for each fixed i, as the one that maximizes the clusterization level of the PAs in the set Θ_i.* By clusterization we mean that the angles between similar and different local subspaces are well separated. In the ideal case (no noise and perfectly orthogonal local subspaces) the PAs would cluster around 0 and $\pi/2$. In real cases the PAs are not perfectly clustered, however, it is possible to evaluate the clusterization level for each Θ_i^r and select the one with the highest clusterization level for each i. We propose to measure the clusterization of each Θ_i^r by using a function inspired by the Linear Discriminant Analysis, we call it Principal Angles Clusterization (PAC):

$$\mathrm{PAC}(\Theta_i^r) = \frac{(\mu_a - \mu_{\mathrm{PAC}})^2 + (\mu_b - \mu_{\mathrm{PAC}})^2}{\sigma_a^{\gamma(\sigma_a)} + \sigma_b^{\gamma(\sigma_b)}} \tag{6}$$

where μ_{PAC} is the center of Θ_i^r computed as the mean of the $\mathcal{P}$ largest and smallest angles, μ_a, σ_a and μ_b, σ_b are the arithmetic means and the standard deviations of the PAs that are above and below μ_{PAC}, respectively. Our tests have shown that $\mathcal{P} = 25\%$ of the Θ_i^r gives a μ_{PAC} that is robust with respect to the presence of outliers (due to oscillations of the PAs). Note that μ_{PAC} is not computed as the mean of all the PAs to avoid biases due to the unbalanced number of representatives of one or the other class. In our experiments r goes from 2 to $8N$.

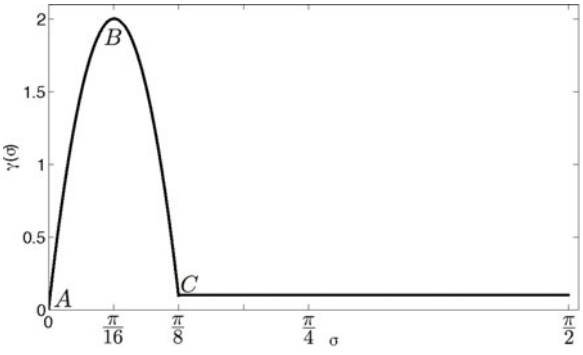


Fig. 2. $\gamma(\sigma)$ function used in the PAC formula

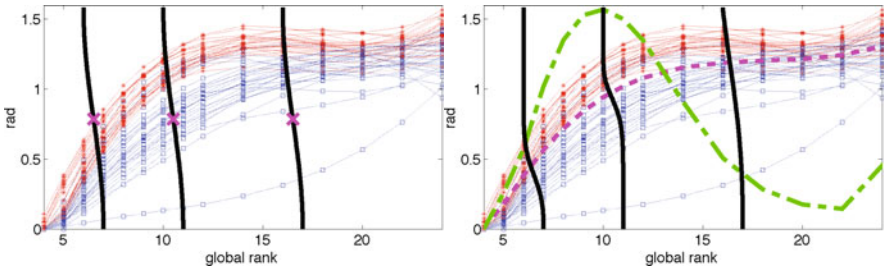
An important component of the formula is the functional exponent $\gamma(\sigma)$. If we used $\gamma(\sigma) \equiv 2$, as in the LDA formulation, the maximum of the PAC function would always be in the extremes of its domain. In fact, it was explained in section 2.2 that when $r \simeq 2$ the PAs tend to cluster around 0, hence the tiny values of the σ's that appear in the denominator of Eq. (6) would boost the PAC value, despite the fact that the μ's are very close to each other. At the other extreme, when $r \simeq r_{\max}$, the μ's increase and become well separated even though the two classes partially overlap. However, as the σ's remain smaller than 1, the global effect would be a magnification of the numerator, boosting again the PAC value. Hence, it is necessary to use a variable exponent that takes small values at the extremes while approaching to 2 for middle values.

A simple function that complies with these requirements is the following:

$$\gamma(\sigma) = \begin{cases} a_1\sigma^2 + a_2\sigma & \text{if } \sigma \leq \pi/8 \\ 0.1 & \text{if } \sigma > \pi/8 \end{cases} \tag{7}$$

The numerical coefficients a_1 and a_2 are not chosen after a tuning procedure but are determined through the following reasoning. Assuming an average case with PAs uniformly distributed, $\mu_{\text{PAC}} = \pi/4$, $\mu_a = 3\pi/8$ while $\mu_b = \pi/8$. Therefore, the upper bound of $\sigma_a, \sigma_b < \pi/8$. The numerical coefficients $a_1 = -50.63$ and $a_2 = 20.13$ define a function that fulfills the previous request making the parabola passing through the points $A \equiv (0, 0)$, $B \equiv (\pi/16, 2)$ and $C \equiv (\pi/8, 0.1)$, as shown in Fig. 2. When σ's $> \pi/8$ the angles are excessively spread and the two classes are likely to overlap. For this reason we maintain $\gamma(\sigma) \equiv 0.1$.

Summarizing, we select for the next step the set of angles in the Θ_i^r with the highest PAC value for each i. Note that the selected rank may be different for each Θ_i. This is a new interpretation of the size of the global subspace: we are not estimating the rank of $\mathbf{W}$, but we are identifying the "most expressive" dimension for each set Θ_i in terms of clusterization level. An example of the PAC function applied to a Θ_i can be seen in Fig. 3b.



(a) $\cos^2$ functions (black) (b) CbA functions (black)

Fig. 3. Example of a random subset of the PAs of Θ_M (largest PAs, 3 rigid independent motions, hence maximum rank 12). PAs between similar subspaces are represented with blue squares, PAs between different subspaces are represented with red asterisks. The affinity functions computed at the rank $r = 6, 10, 16$, appear in black. In Fig. 3a the inflection point of the function is denoted with a magenta cross. In Fig. 3b the magenta dotted line is μ_{PAC} (which for every r it is also the inflection point of the CbA function), the green line-dot-line is the value of the PAC function.

2.4 Sum of Clusterization-Based Affinity (SCbA)

Another fundamental step of manifold clustering based algorithms is to compare subspaces through an affinity measure (as a measure of (dis)similarity). In the literature it is possible to find many affinity measures with different characteristics. A discussion of different affinity measures can be found in [5].

All affinity measures applied to PAs share a common assumption: the angles between similar subspaces are always close to zero, and the angles between different subspaces are always close to $\pi/2$. None of them takes into account that the recursive definition of the PAs tends to force the angle between two subspaces to increase as we move from Θ_i^r to Θ_{i+1}^r. Moreover, none of them takes into account that the angles in a given Θ_i tend to increase when r increases, as explained in section 2.2.

In the example of Fig. 3a we have randomly plotted some PAs of Θ_M of the sequence 1R2RCR (Hopkins155 database). In black it is possible to see the $\cos^2$ function. The $\cos^2$ function always has the same shape and the inflection point (magenta cross) is always in the same position, regardless of the rank to which it is applied. As a consequence of this *rigidity*, if the estimated rank is $r = 6$ all of the PAs have an affinity value that falls before the inflection point. Opposite cases are when $r = 10$ and $r = 16$, in which most of the PAs have an affinity value after the inflection point. Therefore, the $\cos^2$ function, as well as any other rigid function, is very sensitive to the rank estimation.

The affinity measure that we propose is able to adapt itself to the distribution of the PAs in Θ_i^r, so that it minimizes the negative effects of a wrong rank estimation and it emphasizes the difference between similar and different subspaces. We define the not normalized Clustering-based Affinity ($\overline{\text{CbA}}$) between two generic subspaces S_j, S_l, for $j, l = 1, \ldots, P$, for a given Θ_i^r as the function $\overline{\text{CbA}} : \Theta_i^r \to \mathbb{R}^+$,

$$\overline{\text{CbA}}(\theta_i^r(S_j, S_l)) = \exp\left(-\frac{\beta-1}{\beta}\left(\frac{\theta_i^r(S_j, S_l)}{\alpha}\right)^{\beta}\right) \tag{8}$$

where θ_i^r is the i^{th} principal angle computed at the rank r. α and β are the two positive parameters ($\alpha > 0$, $\beta \geq 2$) that allow the function to change in relation to the distribution of the PAs. We can now define the normalized Clustering-based Affinity (CbA) as follows:

$$\text{CbA}(\theta_i^r(S_j, S_l)) = \frac{\overline{\text{CbA}}(\theta_i^r(S_j, S_l)) - \min(\overline{\text{CbA}})}{\max(\overline{\text{CbA}}) - \min(\overline{\text{CbA}})} \tag{9}$$

The arrangement of the parameters of Eq. (8) has been chosen so that CbA has a negative first derivative over all its domain, while its second derivative is negative for $\theta < \alpha$, positive for $\theta > \alpha$ and equal to zero for $\theta = \alpha$. We propose to set $\alpha = \mu_{\text{PAC}}$ so that the inflexion point occurs at the estimated center of the distribution. In this way the function is always stretched or compressed in order to fit the distribution of the PAs. The β parameter is used in order to emphasize the differences between similar and different subspaces in an automatic fashion. In fact, β controls the slope of the function: the higher the β the steeper the slope. We would like an affinity function with a steep slope when the PAs are well clustered and a more gentle slope when the clusterization is not clear. A natural candidate for β is $\beta = \text{PAC}(\Theta_i^r) \cdot \mathcal{F}$, as the PAC function gives a measure of how well clustered the two groups are and how far away the two centroids are. $\mathcal{F}$ is a constant, a boosting factor, that we use in order to give more or less importance to β. In all of our experiments we have used $\mathcal{F} = 5$ which has empirically shown to be a suitable factor.

In Fig. 3b we plot three CbA functions applied to different ranks r within the set Θ_M. In this picture it is possible to appreciate that, thanks to the parameter α, the inflection point changes so that it always corresponds to the μ_{PAC} value, hence minimizing the effect of possible errors in the choice of the rank r. Moreover, thanks to the parameter β the slope of CbA changes depending on how well the small angles are separated from the large angles.

The final affinity between two subspaces is defined as the normalized weighted Sum of CbA (SCbA):

$$\text{SCbA}(S_j, S_l) = \frac{\sum_{i=1}^{M} \text{CbA}(\theta_i^r(S_j, S_l))\text{PAC}(\Theta_i^r)}{\sum_{i=1}^{M} \text{PAC}(\Theta_i^r)} \tag{10}$$

M being the minimum size between subspaces S_j and S_l. In this work we have not investigated the estimation of the local subspace size which was fixed to 4. Note that by weighting the CbA values by the PAC function we give more importance to Θ_i^r where the angles between similar and different subspaces are well separated.

SCbA respects the axioms of an affinity function proposed in [5]:

- **symmetry:** from Eq. (10) we see that $\mathrm{SCbA}(S_j, S_l) = \mathrm{SCbA}(S_l, S_j)$;
- **orthogonality consistency:** given that

$$S_j \perp S_l \iff \theta_i^r(S_j, S_l) = \pi/2 \tag{11}$$

$\forall i = 1, \ldots, M$, from Eq. (9) and (10) it follows that:

$$\mathrm{SCbA}(S_j, S_l) = 0 \tag{12}$$

- **inclusion consistency:** given that

$$S_j \subseteq S_l \iff \theta_i^r(S_j, S_l) = 0 \tag{13}$$

$\forall i = 1, \ldots, M$, from Eq. (10) it follows that:

$$\mathrm{SCbA}(S_j, S_l) = 1 \tag{14}$$

2.5 Summary of Our Proposal

In this section we summarize our proposal: LSA+PAC+SCbA.

1. Build a trajectory matrix $\mathbf{W}$;
2. for $r = 2$ to $r_{\max}$ (in our tests $r_{\max} = 8N$)
   (a) project every trajectory, which can be seen as a vector in $\mathbb{R}^{2F}$, onto an $\mathbb{R}^r$ unit sphere by singular value decomposition (SVD) and truncation to the first r components of the right singular vectors;
   (b) exploiting the fact that in the new space (global subspace) most points and their closest neighbours lie in the same subspace, compute by SVD the local subspaces generated by each trajectory and its nearest neighbours (NNs);
   (c) compute PAs between all of the subspaces;
3. smooth the PAs;
4. apply PAC to find the best r for each Θ_i $(i = 1 \ldots M)$;
5. apply SCbA to build the affinity matrix $\mathbf{A}$;
6. cluster $\mathbf{A}$ by K-means in order to have the final motion segmentation.

More details can be found in the source code available at: `http://eia.udg.es/~zappella`.

3 Experiments

We tested our proposal on the 155 real sequences of the Hopkins155 database and we compared our performances with: LSA + MS (with $k = 10^{-7.5}$, best k value as explained in [12]), ELSA EMS+ (results extracted using available

Table 1. State of the art comparison. Misclassification rates on the Hopkins155 database. In brackets the number of sequences for each type of video. NA stands for value not available.

2 Motions	Checkboards(78)		Articulated(11)		Traffic(31)		All types(120)	
Method	% Avg	% Std	% Avg	% Std	% Avg	% Std	% Avg	% Std
LSA + MS	5.15	9.61	3.65	4.29	4.95	8.66	4.96	8.96
LSA 4N	2.57	6.79	4.10	6.47	5.43	11.17	3.45	8.14
ELSA EMS+	2.20	7.19	2.32	3.87	5.58	10.89	3.08	8.17
ALC	1.49	4.58	10.70	15.00	1.75	1.83	2.40	6.35
MCS + NSI	3.75	7.89	8.05	8.51	1.69	7.00	3.61	7.84
SSC	1.12	NA	0.62	NA	0.02	NA	0.82	NA
Our Proposal	**1.00**	5.64	**1.75**	3.13	**0.57**	1.06	**0.96**	4.67
3 Motions	Checkboards(26)		Articulated(2)		Traffic(7)		All types(35)	
Method	% Avg	% Std	% Avg	% Std	% Avg	% Std	% Avg	% Std
LSA + MS	19.09	13.02	9.57	13.54	16.06	5.72	17.94	11.91
LSA 4N	5.70	10.89	7.25	9.30	25.30	19.05	9.71	14.71
ELSA EMS+	8.76	15.18	6.38	9.03	6.354	12.36	8.15	14.14
ALC	5.00	9.14	21.08	28.87	8.86	13.16	6.69	11.48
MCS+NSI	2.29	5.73	6.38	9.03	1.67	1.51	2.87	5.28
SSC	2.97	NA	1.42	NA	0.58	NA	2.45	NA
Our Proposal	**2.41**	8.05	**3.72**	5.26	**1.11**	1.87	**2.22**	7.03

code [12]), LSA 4N (results taken from [5]), MSC+NSI (results taken from [5]), ALC (results taken from [5]), and SSC (results taken from [3]).

Table 1 shows the average misclassification rates and the standard deviations of each method. The misclassification rates are presented for each type of video sequence (checkboards, articulated and traffic). Firstly, it is possible to see that our proposal outperforms every LSA-based technique proving that our method improves the weaknesses of LSA. Also when the other techniques are taken into account, our proposal has, together with SSC, the lowest misclassification rates both with 2 and 3 motions (the average misclassification rate of our proposal on the whole Hopkins155 database is of 1.25%). However, we would like to remark that for our algorithm the only two free parameters, ($\mathcal{P}$ and $\mathcal{F}$) were fixed for the whole database whereas it is not clear from [3] whether the results of SSC where obtained with a fixed set of parameters or each sequence required a different set.

In Fig. 4 the histogram of the misclassification rates of our proposal is presented. The majority of the sequences, 134, has a misclassification rate smaller than 1%, and the total number of sequences with a misclassification rate below 5% is 145. The median misclassification of every group is always 0% with the exception of the articulated with 3 motions group where the median is equal to the mean (due to the presence in this group of only 2 sequences).

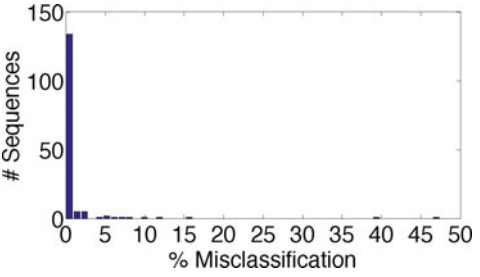


Fig. 4. Histogram of the misclassification rate of our proposal

As far as the computational time is concerned, the bottle neck or our method is the interpolation process of all the PAs. In fact, on the whole Hopkins155 database our proposal required 147600 seconds, of which 143775 were spent for the interpolation (Matlab implementation on Quad-Core AMD @ 2.4GHz, with 16 GB RAM).

In order to verify how our proposal performs on a different database, we tested it on synthetic sequences with 2, 3, 4 and 5 rigid and independent motions (10 different sequences for each number of motions) and an increasing noise level. Specifically, each sequence is composed of 50 frames, with rigidly rotating and translating cubes. Each cube has 56 tracked features. Then we created 2 additional databases adding noise with standard deviations of 0.5 and 1 pixel to the tracked feature positions. In total we used 150 synthetic sequences. The misclassification rates are shown in table 2. All the misclassification rates are smaller than 1%. For a given number of motions the misclassification remains rather stable even when the noise level increases. Moreover, the behaviour of our proposal even with 4 and 5 motions (more than the motions in the Hopkins155 database) is very satisfactory.

Table 2. Misclassification rates on synthetic sequences with 2, 3, 4 and 5 motions and increasing noise level. In brackets the number of sequences for each type of video.

Motions	2(10)		3(10)		4(10)		5(10)	
Our Proposal	% Avg	% Std	% Avg	% Std	% Avg	% Std	% Avg	% Std
$\sigma_{\text{noise}} = 0$	0	0.0	0.24	0.31	0.36	0.35	0.68	0.39
$\sigma_{\text{noise}} = 0.5$	0.09	0.28	0.12	0.25	0.31	0.22	0.75	0.43
$\sigma_{\text{noise}} = 1$	0.27	0.60	0.24	0.31	0.31	0.22	0.75	0.36

4 Conclusions and Perspectives

We presented two improvements for motion segmentation based on manifold clustering. The first improvement is a new way of selecting the global subspace size based on the analysis of the principal angles clusterization, such that the selected size is the one where the principal angles between similar and different

subspaces are best separated. The second improvement is a new affinity measure that is automatically able to adapt itself in order to fit the distribution of the principal angles. The major achievement of this measure is that it can deal with every distribution of principal angles minimizing the effect of an erroneous rank estimation of **W** while maximising the distance between similar and different local subspaces. The results of our experiments show that, even without changing the value of the only two free parameters that we have, the misclassification rates of our proposal are among the lowest in the literature.

Future works should aim to reduce the computational time of the algorithm by adopting other ways for reducing the principal angles oscillations. Moreover, better segmentations could be achieved by extending our algorithm to the estimation of the local subspaces size.

Acknowledgement. This work has been supported by the Spanish Ministry of Science projects DPI2007-66796-C03-02 and DPI2008-06548-C03-03/DPI. L. Zappella is supported by the Catalan government scholarship 2009FI_B1 00068. E. Provenzi acknowledges the Ramón y Cajal fellowship by Ministerio de Ciencia y Tecnología de España.

References

1. Absil, P.A., Edelman, A., Koev, P.: On the largest principal angle between random subspaces. Linear Algebra and its Applications 414, 288–294 (2006)
2. Björck, A., Golub, G.H.: Numerical methods for computing angles between linear subspaces. Mathematics of Computation 27, 579–594 (1973)
3. Elhamifar, E., Vidal, R.: Sparse subspace clustering. In: Proc. CVPR, pp. 2790–2797. IEEE, Los Alamitos (2009)
4. Pinho da Silva, N., Costeira, J.P.: Subspace segmentation with outliers: a grassmannian approach to the maximum consensus subspace. In: Proc. CVPR, pp. 1–6. IEEE, Los Alamitos (2008)
5. Pinho da Silva, N., Costeira, J.c.P.: The normalized subspace inclusion: Robust clustering of motion subspaces. In: IEEE I. Conf. Comp. Vis., pp. 1444–1450 (2009)
6. Lauer, F., Schnrr, C.: Spectral clustering of linear subspaces for motion segmentation. In: IEEE I. Conf. Comp. Vis., pp. 678–685 (2009)
7. Rao, S.R., Tron, R., Vidal, R., Ma, Y.: Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories. In: Proc. CVPR, pp. 1–8. IEEE, Los Alamitos (2008)
8. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. Int. J. Comput. Vision 9, 137–154 (1992)
9. Tron, R., Vidal, R.: A benchmark for the comparison of 3-d motion segmentation algorithms. In: Proc. CVPR, pp. 1–8. IEEE, Los Alamitos (2007)
10. Vidal, R., Hartley, R.: Motion segmentation with missing data using powerfactorization and gpca. In: Proc. CVPR IEEE, vol. 2, pp. 310–316 (2004)
11. Yan, J., Pollefeys, M.: A factorization-based approach for articulated nonrigid shape, motion and kinematic chain recovery from video. IEEE Trans. Pattern Anal. Machine Intell. 30, 865–877 (2008)
12. Zappella, L., Lladö, X., Provenzi, E., Salvi, J.: Enhanced Local Subspace Affinity for Feature-Based Motion Segmentation. Pattern Recognition 44, 454–470 (2011)

Real-Time Detection of Small Surface Objects Using Weather Effects

Baojun Qi, Tao Wu, Hangen He, and Tingbo Hu

Institute of Automation, College of Mechatronics Engineering and Automation, National University of Defense Technology, Changsha 410073, P.R. China

Abstract. Small surface objects, usually containing important information, are difficult to be identified under realistic atmospheric conditions because of weather degraded image features. This paper describes a novel algorithm to overcome the problem, using depth-aware analysis. Because objects-participating local patches always contain low intensities in at least one color channel, we detect suspicious small surface objects using the dark channel prior. Then, we estimate the approximate depth map of maritime scenes from a single image, based on the theory of perspective projection. Finally, using the estimated depth map and the atmospheric scattering model, we design spatial-variant thresholds to identify small surface objects from noisy backgrounds, without contrast enhancement. Experiments show that the proposed method has real-time implementation, and it can outperform the state-of-the-art algorithms on the detection of distant small surface objects with only a few pixels.

1 Introduction

Small surface objects always contain important information. Radar system, however, is ineffective against small objects close to the vessel [6]. In addition, for objects without hot parts present (e.g., buoys, surface rocks), near infrared images get unsuitable for detection either [16]. Therefore, detecting small surface objects from visual images is highly desired.

There has been growing interest in surface targets identification, as most standard computer vision algorithms for traffic scene are ineffective for maritime conditions [2, 16]. To avoid collisions of surface vessels, Sanderson et al. have proposed some maritime targets identification algorithms [16,17], using statistical characteristics of the sea and motions of the targets. Sullivan et al. [18] use an optimal trade-off MACH filter to detect vessels from maritime surveillance videos. But the mach filter for various targets needs to be trained beforehand. Gupta et al. [6] provides an approach for maritime objects recognition through case-based statistical relational learning.

Algorithms, described above, require robust detection of image features. Under realistic atmospheric conditions, however, images of outdoor scenes are usually affected by scattering medium [11,12] (e.g., small aerosols, water-droplets). The affection, usually leading to contrast lost or color infidelity, increases exponentially with the distances of scene points from the sensor [13]. To eliminate

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 27–38, 2011.

the depth-dependent degradation, many haze removal algorithms are provided to preprocess the hazy images [4,7,13,19], which may lead to time-consuming detection algorithms. Thus, we intent to use the spatial-variant weather effects for surface objects detection without preprocessing hazy images.

In this paper, we describe an algorithm for small surface objects detection, using depth-aware analysis of image features by the atmospheric scattering model. Our purpose is giving early warning sources of information to avoid collisions or terrorist attacks. The main contributions of the paper are as follows. Firstly, we locate suspicious small surface objects with the dark channel prior, proposed by He et al. [7] from the statistics of hazy free outdoor images. The prior is based on the observation that objects-participating local patches have very low intensities in at least on color channel because of shadows, high-colored or dark objects. Secondly, we detect the horizon with dyadic cubic spline wavelet transforms [10], and estimate the scaled depth map of the sea surface based on the theory of perspective projection [5]. Lastly, using the atmospheric scattering model and the estimated depth map, we obtain spatial-variant thresholds and real-time detection of small surface objects, without hazy removal manipulations beforehand. Note that, our algorithm can distinguish objects from moving waves effectively, and it can outperform the state-of-the-art techniques [1,9,15] for detecting small objects with only a few pixels. In addition, when sequential information is used for the estimation of the depth map, the computational complexity for detecting objects from an $m \times n$ RGB image is $7mn$ times comparisons.

Our approach does have limitations. It becomes invalid for objects that are inherently similar to the sea surface and without any shadows casted on. However, we believe that the techniques used in this paper (e.g., designing depth-aware thresholds according to the weather effects, detecting small objects by local filters) can provide useful indications for other computer vision algorithms.

2 Weather Effects on Vision

In participating medium, light reflected from an object gets scattered by atmospheric particles that have significant size and concentration, leading to weather degraded images [19]. As the atmosphere on the sea always has high quantities of hygroscopic particles (e.g., sea salt), weather effects must be considered on surface objects detection under realistic weather conditions.

2.1 Atmospheric Scattering Model

In computer vision, the widely used model to describe scattering effects is:

$$E(x) = R(x)t(x) + E_\infty [1 - t(x)] \tag{1}$$

where x is the scene point, E is its observed intensity, R is its scene radiance, E_∞ is the horizon brightness, and t is its transmission describing the percent of the light reaching the observer. For homogenous medium, the transmission can be expressed as:

$$t(x) = e^{-\beta d(x)} \tag{2}$$

Here, d is the depth of the scene point from the observer. β is the scattering coefficient, related to the meteorology visibility V as $\beta = \frac{3.912}{V}$ for homogeneous medium. Generally, the first term on the right hand side of Eq. (1) is called *attenuation*, and the second term is called *airlight*, denoted as A in the paper.

2.2 Degradation Analysis

According to Eq. (1), the observed contrast for two adjacent points i and j is:

$$\left|\frac{E_i - E_j}{E_i + E_j}\right| = \frac{|R_i - R_j|}{(R_i + R_j) + 2E_\infty(e^{\beta d} - 1)} \tag{3}$$

This means that the differences of foreground and background degrade exponentially with the depths of scene points. Thus, many haze removal algorithms have been proposed recently for robust detection of image features [4,7,12,13,14,19].

Real-time vision systems always require single image based contrast enhancement techniques. As depth estimation from a single image is an ill-posed problem, many algorithms estimate the airlight based on some kind of prior first and then obtain the depth map [4,7,19]. The dark channel prior [7] indicates that the dark channel R^{dark} tends to be zero for most object-participating local patches,

$$R^{dark}(x) = \min_{x \in \Omega(x)} \left(\min_{c \in \{r,g,b\}} (R^c(x)) \right) \tag{4}$$

where $\Omega(x)$ is the local patch of x. If we assume the airlight of the points in a local patch to be identical, it can be estimated by $A = \min_{\Omega} \left(\min_{c} (E^c) \right) - t(x) R^{dark}$. In addition, the scaled depths of the objects-participating patches can be estimated according to Eq. (1) and Eq. (4) as:

$$\beta d = -\ln t = -\ln \left(1 - \min_{\Omega} \left(\min_{c} \frac{E^c}{E^c_\infty} \right) \right) \tag{5}$$

That is, we can estimate the degradation of hazy images and the 3D structure of the scene using the dark channel prior.

3 Detection Using Weather Effects

3.1 Dark Channel Prior for Surface Objects

The rationality of the dark channel prior has been verified by He et al. [7], based on the statistical analysis of large numbers of haze-free outdoor images. The prior has made great success in haze removal [7] as well as the 3D structure estimation from a single image [3,7]. However, the prior becomes invalid for maritime scenes which have large regions similar to the atmospheric light. Therefore, we directly detect surface objects without applying haze removal algorithms.

Figure 1 shows an example of the dark channel image of a sea surface image, obtained by local minimum filters with 5×5 rectangular kernels [8]. Note that, the dark channel image has the following characteristics. First, small surface

Fig. 1. Left: input image. Right: the dark channel image with 5×5 rectangular kernels.

objects with only a few pixels are enhanced and enlarged in the dark channel image. Second, for moving waves, only those near the sensor have low intensity, which may interfere with distant small objects detection. However, weather effects analysis can help us remove those waves as they are not dark enough if considering the additive airlight. Last, because the mountains are farther than surface objects, the dark channel of the mountains is with higher intensities according to Eq. (1). The observation can distinguish surface objects from the mountains far away.

In the rest of the paper, we first estimate the depth map of the sea surface. And then, we compute the scattering effects by the estimated depth map, followed by surface objects detection with spatial-variant thresholds.

3.2 Depth Map from a Single Image

Our purpose here is estimating each pixel's rough depth with respect to (w.r.t.) its row index. Based on the theory of the perspective projection, we have the following results.

Proposition 1. *Denote the pixel's row index as v. Assuming pixels in the same row with an identical depth d, then the depths of other pixels can be obtained by:*

$$d = d_0 \frac{v_0 - c}{v_0 - h} \cdot \frac{v - h}{v - c} \tag{6}$$

here, c is the horizon, $v \neq c$, h is the height of the image, and d_0 is the depth of a special pixel with $v = v_0$.

Proof. According to the theory of the perspective projection, the homogeneous coordinate of a point $U_{img} = (u, v, 1)^T$ in image plane can be expressed with:

$$Z \cdot U_{img} = M_{3\times4} X_w \tag{7}$$

where $X_w = (x_w, y_w, z_w)^T$ is the homogeneous coordinate in world, and $M_{3\times4}$ is the projection matrix. We only consider the depth map in the horizontal plane,

and assume the pixels in the same row have the same depth. Then, $x_w = 0$ and $y_w = 0$ in Eq. (7). A scene point's depth w.r.t. its row index can be derived as:

$$v = \frac{a}{z_w + b} + c' \tag{8}$$

where $a = \frac{m_{24}}{m_{33}} - \frac{m_{23}m_{34}}{m_{33}^2}$, $b = \frac{m_{34}}{m_{33}}$, and $c' = \frac{m_{23}}{m_{33}}$. We assume the size of the image is $h \times w$. The parameters in Eq. (8) are estimated as follows.

(1) Let $z_w = \infty$, then $v = c'$. That is, c' is the vanishing line or the horizon of the maritime images, denoted as c in the following.
(2) Assuming the row index h having the minimum depth d_{min}, we obtain $h = \frac{a}{d_{min}+b} + c$. Thus, $a = (h - c)(d_{min} + b)$.

Assume that we have estimated the depth z_2 of some pixel, then the depths of other pixels can be derived from Eq. (8) as:

$$\frac{z_1}{z_2} = \frac{v_2 - c}{v_1 - c} \cdot \frac{a - b(v_1 - c)}{a - b(v_2 - c)} \tag{9}$$

Substituting estimated a into Eq. (9) and assuming $d_{min} \approx 0$, we obtain

$$\frac{z_1}{z_2} = \frac{v_2 - c}{v_1 - c} \cdot \frac{v_1 - h}{v_2 - h} \tag{10}$$

Denote the depth as d instead of z in Eq. (10), we finally get

$$d = d_0 \frac{v_0 - c}{v_0 - h} \cdot \frac{v - h}{v - c} \tag{11}$$

where v and d are the row index and the depth of a scene point respectively, c is the horizon, and d_0 is the depth of a reference point. Equation (6) indicates that once the horizon is estimated, the depth map of the maritime scenes can be computed from a reference pixel with known depth. □

Corollary 1. *When misestimate of c occurs, images with smaller c have more robust estimations of depths.*

Proof. Assume the estimated horizon to be $c' = c + \varepsilon$. According to Eq. (11), the estimated error is:

$$\left| \frac{d'}{d} - 1 \right| = \left| \frac{\varepsilon(v_0 - v)}{(v - c - \varepsilon)(v_0 - c)} \right| \tag{12}$$

Note that, the points of the sea surface usually satisfy $v > c$ in the image plane. When the detection error ε of the horizon cannot be avoided, Equation (12) shows that larger $|v_0 - c|$ implies more robust estimation of d. As the reference point v_0 is on the sea surface, larger $|v_0 - c|$ indicates smaller c. □

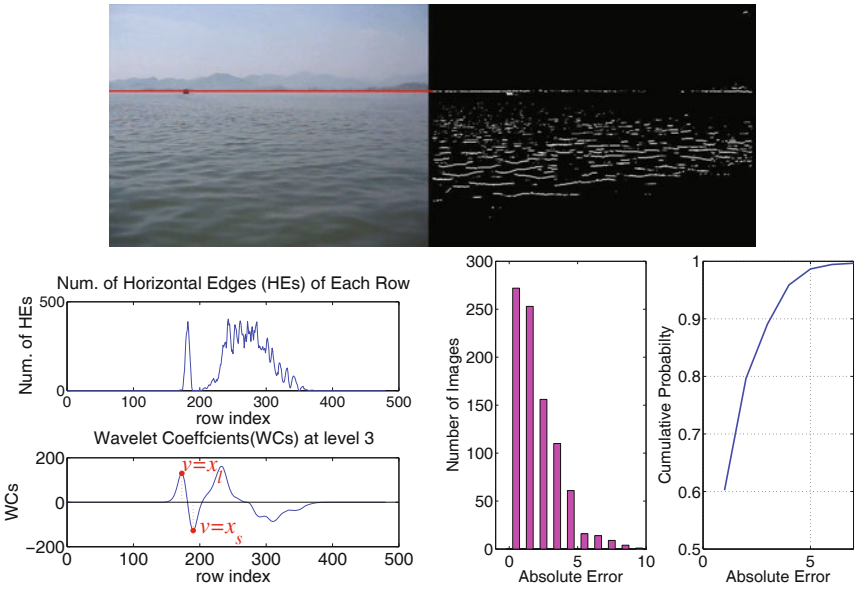


Fig. 2. Detection of the horizon. Topleft: detection results of the horizon. Topright: detection of the horizontal edges. Bottomleft: the number of detected horizontal edges for each row and its corresponding wavelet coefficients at level 3. Bottomright: histogram of the absolute detection error and its corresponding cumulative distributions.

3.3 Detecting the Horizon

Note that, estimating the depth map needs detection of the horizon. Additionally, surface objects are usually around the horizon due to the perspective projection. Therefore, this subsection mainly discuss how to detect the horizon.

To obtain robust detection, we apply the wavelet transforms based singularity analysis techniques. The wavelet we used is the orthogonal cubic dyadic spline wavelet transforms, whose coefficients for decomposition are given in Table 1. The advantage of the method is that it can detect various types of edges through multi-resolution analysis [10]. Additionally, the wavelet is translation invariant. If the low-pass filters and the high-pass filters for decomposition are $\{h_m\}$ and $\{g_m\}$ respectively, the wavelet coefficients at level j for input signal a^0 are:

$$d_n^j = (a^{j-1} * g^{j-1})_n = \sum_m g_m a_{n-2^{j-1}m}^{j-1} \tag{13}$$

$$a_n^j = (a^{j-1} * h^{j-1})_n = \sum_m h_m a_{n-2^{j-1}m}^{j-1} \tag{14}$$

For an input image, we detect the horizontal edges by labeling pixels with larger magnitudes of wavelet coefficients than some threshold. Then, we compute each row's number of edges and construct a 1D vector. Moving waves, shown as the

Table 1. Coefficients of orthogonal cubic dyadic spline wavelet. h_n, g_n are the low-pass filters and the high-pass filters respectively for decomposition, where $h_{-n}=h_n$, $g_{-n}=-g_n$ for $n \leq 4$, and $h_n=0$, $g_n=0$ for $|n|>4$.

n	0	1	2	3	4
$h_n/\sqrt{2}$	0.3750	0.2500	0.0625	0	0
$g_n/\sqrt{2}$	0	0.59261	0.10872	0.01643	0.00008

top right of Fig.2, are hindrances to the horizon detection. However, we discover that there is a peak around the horizon and the horizontal edges of waves only existing nearby. Thus, we analyze the wavelet coefficients of the 1D signal at large scale space, e.g. Level 3. We detect points with the smallest wavelet coefficients (denoted as $v = x_s$), and points with the largest wavelet coefficients ($v = x_l$) between $v=1$ and $v=x_s$. The horizon is assumed to be $\frac{x_l+x_s}{2}$ (shown in Fig.2). For randomly selected 600 maritime images which have land on the horizons or have sea/sky horizons, we manually label the ground truth horizons. Our algorithm is tested on the data set and obtains a performance with 0.5 pixel mean error and 2.7 pixels stand deviation. Figure 2 shows the histogram of the absolute detection error and its corresponding cumulative distributions.

3.4 Spatial-Variant Thresholds

According to the subsection 3.2, the depth-map of the sea surface can be computed with the horizon and some reference point with known depth. In this subsection, we firstly discuss how to select the reference point and estimate its scaled depth. Then, we describe the method for surface objects detection from the dark channel image, using depth-variant thresholds which are computed from the estimated depth map in subsection 3.2 according to the scattering model.

Selecting the Reference Point. As discussed in Section 3.2, the depth of the reference point should be computable. Thus, the reference point must be selected from the objects-participating local patches which satisfy the dark channel prior. When multiple objects share the same row index, we choose the furthermost object's location as the reference point. The procedure is as follows.

(1) Compute the dark channel image dch for the input image img, the local minimum operators we used is $m \times n$ rectangular kernels;
(2) For dch, compute the histogram of the horizon centered region R_{int}, denoted as H_{dch}(Left of Fig. 3). Because objects-participating local patches tend to have pixels with lower intensities, we estimate the reference point's intensity A_0 as follows.
   - Search the first i satisfying $H_{dch}(i)<\varepsilon$ (e.g. $\varepsilon=1$)from $i=\arg\max_j\{H_{dch}(j)\}$ to $i=0$;
   - For some δ, look for the first ii satisfying $H_{dch}(ii)>\delta$ from $ii=i$ to $ii=0$, which is just the A_0 we needed.

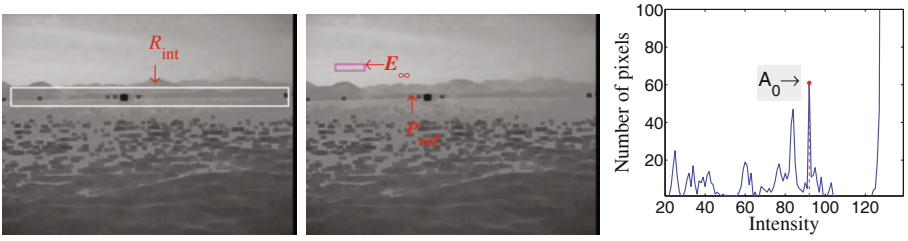


Fig. 3. Selecting the reference point P_{ref} from the dark channel image. Left: region of interest R_{int} for selecting P_{ref}, bounded by the white rectangle. Middle: Estimation of E_∞ and the local patch contains P_{ref}. Right: the estimated airlight of P_{ref} from the histogram of R_{int}.

(3) As only the row index and the depth of the reference point are concerned, we label the pixels with intensities A_0 and select the one with largest row index v_0 as the reference point. Considering the translation of v_0 caused by local minimum filters, we adjust v_0 to $v_0 \leftarrow v_0 - \frac{m}{2}$.

Then we can compute the depth of the reference point from Eq. (5) as:

$$d_0 = \frac{1}{\beta} \ln \left(\frac{E_\infty}{E_\infty - A_0} \right) \tag{15}$$

Where β is the scattering coefficient, and E_∞ is the horizontal brightness.

Estimating the Airlight. Using Eq. (5), we can estimate the transmission of other pixels as:

$$\frac{\ln t}{\ln t_0} = \frac{d}{d_0} \Longrightarrow t = t_0^{d/d_0} \tag{16}$$

where t_0 is the transmission of the reference point. Additionally, the airlight of other pixels can be estimated by:

$$\frac{A}{A_0} = \frac{E_\infty(1-t)}{E_\infty(1-t_0)} \Longrightarrow A = \frac{A_0}{1-t_0}\left(1 - t_0^{d/d_0}\right) \tag{17}$$

Estimating E_∞ with the method described in [7] and substituting Eq. (10) into Eq. (17), we compute each pixel's airlight w.r.t. its row index v by:

$$A = \frac{A_0}{1-t_0}\left(1 - t_0^{\frac{(v_0-c)(v-h)}{(v_0-h)(v-c)}}\right) \tag{18}$$

where h is the height of the input image, and $t_0 = 1 - \frac{A_0}{E_\infty}$. Figure 3 shows the selection of the reference point and the estimation of E_∞.

Detecting Surface Objects. According to the statistical report of He et al. [7], 90% of the pixels have intensities less than 25 for objects-participating local

Fig. 4. Surface objects detection for surveillance videos of the lake scenes. Top: Original frames of different video clips. The first two images contain objects at various distances on a sunny day, while the last one is captured on a rainy day. Bottom: detection results of our method.

patches in the dark channel image. However, the intensities are often larger due to weather effects (e.g., small aerosols, haze). By quantitative description of the weather effects with the airlight A, we can design depth-variant thresholds by $A+\delta'$ ($\delta'>25$), and detect surface objects from the dark channel image by labeling pixels whose intensities are less than their corresponding thresholds.

4 Experimental Results

In our experiments, the dark channel image is computed using Marcel van Herk's fast local minimum operator [8] with 8×8 rectangular kernels. To obtain robust estimation of the airlight, we use $\hat{c}=c-\sigma$ ($\sigma=20$) instead of c in our experiments, according to the Corollary 1. However, the estimated airlight will shrink due to the adjustment of c. To overcome the problem, we design the thresholds with the estimated airlight plus a constant δ (e.g. $\delta=10$).

Figure 4 shows the results of our algorithm on the surveillance videos of the lake. As can been seen, our algorithm can achieve good performance for objects at various distances on both sunny days and rainy days. Especially, the method works for far objects with only a few pixels (shown in the middle of Fig. 4).

Furthermore, we test our algorithm on various maritime surveillance videos, such as objects with various depths, objects in low contrast regions, and objects in noisy backgrounds. Robust detections are shown in Fig. 5.

To evaluate the performance of our algorithm, we compare our algorithm with other detection methods, which are based on background modeling [20] and saliency analysis [1,9,15]. Different algorithms are tested on large numbers of images captured on the lake and the sea surface. Quantitative evaluation of

Fig. 5. Results on maritime surveillance videos. Top: original. Bottom: results.

the algorithms are analyzed by making the precision/recall (PR) curves, which are obtained by comparing the detection results with the manually labeled objects. The PR curves for different videos are shown in Fig. 6. As can been seen, both background modeling based method and saliency detection based methods have many false detections at large recall values. However, our algorithm make high detection precision for high recall levels (over 90%) as shown in Fig. 6. Thus, our algorithm can outperform the methods described in [1,9,15,20] for small surface objects detection.

Computational Complexity Analysis. In our experiment, computing the dark channel image with Marcel van Herk's method needs $6mn$ times comparisons for an $m \times n$ image, independent of the size of the kernels. The detection of the horizon needs $8mm + 18m$ multiplications and $8mn + 14m$ additives. The estimation of the thresholds needs about $4m$ multiplications and $6m$ additives respectively. Therefore, the computational complexity of our algorithm is linear with the number of pixels. We implement our method in C++ and it takes about 30∼40ms to process a 352×288 image on a PC with 2.6GHZ Intel Pentium-V processor and 1G memory. The algorithm can be accelerated by using sequential information for the estimation of the horizon.

5 Discussions and Conclusions

This paper has described a novel algorithm for small surface objects detection using weather effects. We first compute the depth map of the sea surface from a single image according to the theory of the perspective projection. Then, using the estimated depth map, we compute the spatial-variant airlight by the atmospheric scattering model, and design depth-aware thresholds for surface objects

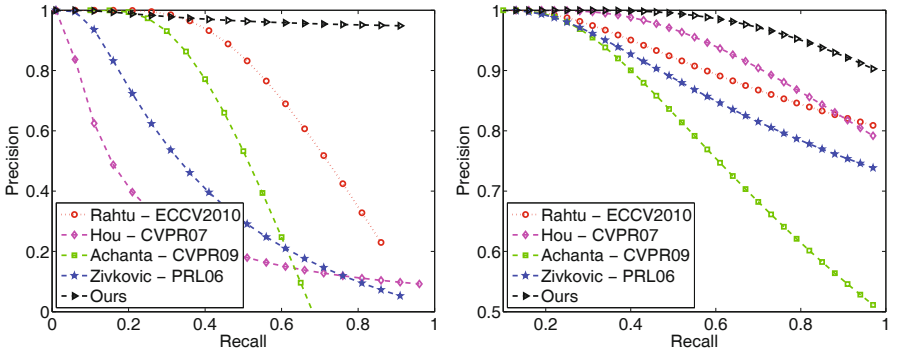


Fig. 6. Comparison of our algorithm with other methods for different videos. Left: results on the surveillance videos with multiple objects at various distances and moving waves presented. Right: results on the maritime surveillance videos with small objects near the horizon.

Fig. 7. Failure case. Left: input image. Right: the dark channel image.

detection from the dark channel image. As shown in Fig. 6, our algorithm has more than 90% true positive rate for high recall values (recall>0.9). Especially, the proposed method outperforms other algorithms described in [1,9,15,20] for detecting faraway objects with only a few pixels.

However, for objects which are inherently similar to the atmospheric light and no shadow is casted on them, our algorithm cannot work because of the invalidation of the dark channel prior, which is just a kind of statistic. We intend to integrate other local descriptors of surface objects with the dark channel prior to improve the performance of our algorithm in the future.

Acknowledgements. The authors would like to thank the reviews for their valuable suggestions. The work is supported by National Natural Science Foundation of China under Grant No.90820302 and No.90820015.

References

1. Achanta, R., Hemami, S., Estrada, F., Süsstrunk, S.: Frequency-tuned salient region detection. In: CVPR (2009)
2. Chan, A.B.: Beyond Dynamic Textures: a Family of Stochastic Dynamical Models for Video with Applications to Computer Vision. PhD thesis, University of California, San Diego (2008)
3. Cozman, F., Krotkov, E.: Depth from scattering. In: CVPR, pp. 801–806 (1997)
4. Fattal, R.: Single image dehazing. In: SIGGRAPH, pp. 1–9 (2008)
5. Forsyth, D.A., Ponce, J.: Computer Vision: A Modern Approach. Prentice-Hall, Englewood Cliffs (2003)
6. Gupta, K.M., Aha, D.W., Moore, P.: Case-based collective inference for maritime object classification. In: McGinty, L., Wilson, D.C. (eds.) ICCBR 2009. LNCS, vol. 5650, pp. 434–449. Springer, Heidelberg (2009)
7. He, K., Sun, J., Tang, X.: Single image haze removal using dark channel prior. In: CVPR, pp. 1956–1963 (2009)
8. Herk, M.V.: A fast algorithm for local minimum and maximum filters on rectangular and octogonal kernels. Pattern Recognition Letters 13, 517–521 (1992)
9. Hou, X., Zhang, L.: Saliency detection: a spectral residual approach. In: CVPR (2007)
10. Mallat, S., Hwang, W.: Singularity detection and processing with wavelets. IEEE Transactions on Information Theory 38, 617–643 (1992)
11. McCartney, E.J.: Optics of the Atmosphere–Scattering by Molecules and Particles. John Wiley and Sons Inc., Chichester (1976)
12. Narasimhan, S.G., Nayar, S.K.: Vision and the atmosphere. International Journal of Computer Vision 48, 233–254 (2002)
13. Narasimhan, S.G., Nayar, S.K.: Contrast restoration of weather degraded images. IEEE Trans. Pattern Anal. Mach. Intell. 25, 713–724 (2003)
14. Nayar, S.K., Narasimhan, S.G.: Vision in bad weather. In: ICCV, pp. 820–827 (1999)
15. Rahtu, E., Kannala, J., Salo, M., Heikkilä, J.: Segmenting salient objects from images and videos. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6315, pp. 366–379. Springer, Heidelberg (2010)
16. Sanderson, J., Teal, M., Ellis, T.: Target identification in complex maritime scenes. In: The Sixth International Conference on Image Processing and its Applications, vol. 2, pp. 463–467 (1997)
17. Smith, A., Teal, M., Voles, P.: The statistical characterization of the sea for the segmentation of maritime images. In: The 4th EC-VIP-MC, vol. 2, pp. 489–494 (2003)
18. Sullivan, M.D.R., Shah, M.: Visual surveillance in maritime port facilities. In: Visual Information Processing XVII, pp. 1–8 (2008)
19. Tan, R.: Visibility in bad weather from a single image. In: CVPR, pp. 1–8 (2008)
20. Zivkovic, Z., van der Heijden, F.: Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern Recognition Letters 27, 773–780 (2006)

Automating Snakes for Multiple Objects Detection

Baidya Nath Saha, Nilanjan Ray, and Hong Zhang

Department of Computing Science, University of Alberta, Edmonton, Canada
{baidya,nray1,hzhang}@ualberta.ca

Abstract. Active contour or snake has emerged as an indispensable interactive image segmentation tool in many applications. However, snake fails to serve many significant image segmentation applications that require complete automation. Here, we present a novel technique to automate snake/active contour for multiple object detection. We first apply a probabilistic quad tree based approximate segmentation technique to find the regions of interest (ROI) in an image, evolve modifed GVF snakes within ROIs and finally classify the snakes into object and non-object classes using boosting. We propose a novel loss function for boosting that is more robust to outliers concerning snake classification and we derive a modified Adaboost algorithm by minimizing the proposed loss function to achieve better classification results. Extensive experiments have been carried out on two datasets: one has importance in oil sand mining industry and the other one is significant in bio-medical engineering. Performances of proposed snake validation have been compared with competitive methods. Results show that proposed algorithm is computationally less expensive and can delineate objects up to 30% more accurately as well as precisely.

1 Introduction

Snake/active contour [1] has made its recognition as an interactive image segmentation tool for the last two decades. However, it is yet to be seen as a completely automated segmentation tool. Snake algorithms consist of three sequential steps: snake initialization, snake evolution and snake validation [2]. For multiple object detection, seeds are chosen inside the objects at the initialization step, then snakes are evolved from those seed points and finally the evolved snakes are passed through a validation procedure to examine whether the snakes delineate the desired objects [2]. Substantial endeavors have taken place on the initialization and evolution steps towards snake automation. Most of the existing initialization algorithms [3] exploit the local maxima or other characteristics of the external energy that help to generate seed points within the objects. However, clutters in the noisy and poorly illuminated images generate considerable amount of seed points and snakes evolved from those seeds do not converge to the object boundaries. This necessitates a good validation scheme after snake evolution. Unfortunately, the validation step has not received much attention till date.

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 39–51, 2011.

Saha *et al.* [2] proposed a snake validation scheme using principal component analysis (PCA). Their method places seeds randomly on the entire image and evolve one snake from each seed. When all snakes converge, a pattern image (an annular band) is formed along each snake contour. Each pattern image is then projected into an already trained PC (principal component) space and PCA reconstruction error is computed. The snakes associated with lower reconstruction errors than a threshold are considered as objects. Pattern images bear information regarding bright-to-dark (or vice-versa) transition across the object contours and show good discrimination capability between object and non-object classes. This validation technique is effective when the gradient strength of object boundaries is considerably high. Besides, throwing a large number of seeds blindly over an entire image might not be feasible for some applications, since the snake evolution can be computationally expensive. Thus, carefully placed seed points are always desirable.

In this paper, we propose a probabilistic quad tree (QT) based snake initialization scheme, which is computationally inexpensive. QT automatically seeks ROIs from an image where the probabilities of locating objects are very high. We throw seeds only within ROIs and evolve one modified Gradient Vector Flow (GVF) snake [4] from each seed. Then we validate each evolved snake to verify whether they belong to object or non-object class. During validation, each snake is passed through a strong classifier formed by Adaboost [5]. We classify snake contours into objects and non-objects based on a set of features and we apply Adaboost for selecting important features. The parameters of the adaboost algorithm are estimated by minimizing an exponential loss function. Here, it is noted that one shortcoming of the exponential loss function associated with Adaboost algorithm is that the penalty that increases exponentially with negative margins incurs high misclassification error rates due to outliers [5]. We propose a novel loss function that incurs smaller penalties in the negative margin, and thus make Adaboost more robust to outliers. Also, we can choose the amount of penalty judiciously from the training set using cross validation. We exploit the advantages of multiple features including region, edge and shape over PCA-based intensity feature proposed earlier [2]. Note that our proposed initialization and validation algorithm could be successfully used as plugins with any existing snake evolution techniques. We have carried out experiments on two real datasets: (a) oil sand mining images [4]: analyzing these images helps to improve the performance of oil sand extraction process and (b) leukocyte images [6]: processing these images helps in the study of inflammation as well as in the design of anti/pro inflammatory drugs. Results illustrate that our proposed algorithm is faster, more reliable and robust than competitive methods.

The organization of this paper is as follows. Section 2 discusses proposed quad tree based snake initialization technique. Section 3 elaborates snake validation using boosting and illustrates proposed regularization into boosting framework. Section 4 demonstrates the performances of proposed techniques and displays comparative analysis of proposed techiques with competitive methods. Section 5 concludes our proposed work. Appendix includes derivation of proposed discrete Adaboost algorithm.

2 Quad Tree Based Snake Initialization

Quad tree [7] based segmentation algorithm receives an image as an input, and then divides it into four adjacent, non-overlapping quadrants if it meets pre-specified criteria. Subsequently each quadrant is divided similarly and the process proceeds iteratively until it fails the pre-defined criteria. Consequently, the algorithm locates objects by smaller rectangular boxes. In our application here, the QT algorithm computes a posterior probability and splits the current region into four quadrants if the value of the posterior probability is between two predetermined thresholds. If the value of the posterior probability is greater than the upper threshold then the region is likely to contain objects; if it is less than the lower threshold then it is likely to contain background. We locate objects by finding homogeneous regions based on local brightness and texture properties. We compute the posterior probability of a region (O) being object/ non-object: $P(O/T, B) \propto P(T/O)P(B/O)P(O)$, where $P(O)$ is the prior probability. $P(T/O)$ and $P(B/O)$ are the likelihood of the region regarding texture and brightness respectively. Proposed probabilistic QT algorithm converges faster and delineates objects more accurately than deterministic quad tree algorithm if a suitable, application specific prior can be chosen. We compute texture energy (T) by the response of Gabor filters [7] and brightness (B) by the maximum singular value decomposition (SVD) [8] of the region. Maximum SVD encodes average brightness and Gabor filter response represents discriminative texture information for the objects. The details of computing posterior probability and two thresholds are mentioned in Section 4.

3 Snake Validation Using Boosting

We compute different features for each converged snake contour, such as, contour shape features (form factor, convexity, extent, modification ratio [9] etc.), regional features (intra and inter class variance, entropy etc.), and edge based features (GICOV [6], gradient strength etc.) for snake validation. We use Adaboost (variant of boosting) for selecting important features. At the training phase, boosting picks only important features for snake validation from a set of features computed on training snake contours and finds the weights associated with those features. We place seeds randomly over the training images and evolve one snake from each seed and classify the snakes as objects manually that converge at object contours found on the ground truth made by the experts; otherwise consider the snakes as non-objects and thus form a training set consisting of both positive (object) and negative (background) samples. The Adaboost algorithm forms a strong classifier by combining a set of weak learners linearly in an iterative manner [5]. We use decision stump (threshold) [5] as weak classifiers. Decision stump is a single level decision tree. Decision stump, $G_j(x)$ for feature f_j is defined as, $G_j(x) = 1$ if $x_j > \theta_j$, otherwise, $G_j(x) = 0$, where θ_j is some feature value of x_j chosen as threshold and $x = [x_1, x_2, x_3, ..., x_j, ...x_n]$ is the feature set. Finding the best decision stump at each stage is similar to learning a node in

a decision tree. We search over all possible features $x = [x_1, x_2, x_3, ..., x_n]$ and for each feature, we search over all possible thresholds θ induced by sorting the observed values of x and pick x_k with θ_k that gives lowest misclassification error among all given features during training. At test phase, proposed QT algorithm discussed in Section 2 locates ROIs (rectangular regions/patches) over the test images where the probability of localizing objects is greater than a predetermined upper threshold. We place seeds only within ROIs and grow one snake from each seed. When all snakes are fully converged, we compute the values of the important features for each snake and multiply them with the weights associated with the features chosen by boosting during training phase and subsequently add them to form a strong classifier, $G(x) = sign(\sum_{m=1}^{M} \alpha_m G_m(x))$, where, α_m is the weight associated with weak classifier $G_m(x)$. If the sign of the response of the strong classifier for a snake contour is positive then it is classified into object class, otherwise it is classified into non-object class.

For classification, Adaboost minimizes an exponential loss function: $L(y,f(x))= exp(-yf(x))$, where y is the response and f is the prediction. The drawback of this exponential loss function is that it incurs substantial misclassification error rate as the penalty increases exponentially for large increasing negative margin due to outliers [5]. To address this problem, we propose a novel loss function: $L(y, f(x)) = exp(-yf(x) + \lambda|y - G(x)|)$, where $\lambda < 0$ and $G(x)$ is the prediction of the weak classifier chosen at the current stage. We have introduced one extra term to the existing exponential loss function that acts as a regularizer. At any boosting iteration, the proposed loss function is the same as the existing loss function if the misclassification error rate at current stage is zero (proposed term vanishes when $\lambda = 0$). The only difference between the proposed and the exponential loss function is that the penalty associated with the proposed loss function is less than that of the exponential one, if the misclassification error rate at current stage is not equal to zero (shown in Fig.1(a) where loss is plotted against a function of the classification margin $y.f$). This modification leads to a low misclassification error rate and it becomes more robust to outliers. One additional advantage of this proposed loss function is that the user can adjust the amount of penalty for negative margins after observing the classifier performance over a training data set. Accordingly, we determine the value of λ through cross validation (λ is a function of k shown in the appendix and the value of k is determined experimentally). We derive a modified Adaboost algorithm by minimizing the proposed loss function (The derivation is shown in Appendix).

Our modified Adaboost finds the feature weight, $\alpha_m = log(k(1-err_m)/err_m)$, $k \geq 1$, where, for the existing Adaboost algorithm the value of k is always 1. This leads to the weights associated with misclassified observations at any stage being k times as much as the existing Adaboost (derivation is shown in the Appendix). The value of k for our modified Adaboost is determined by cross-validation and is discussed in the next section.

Our proposed term in the existing loss function acts as a regularizer in the boosting framework. There are two well known regularized boosting algorithms, ϵ-boosting [5] and l_1- regularized boosting [10] available in the literature. Unlike

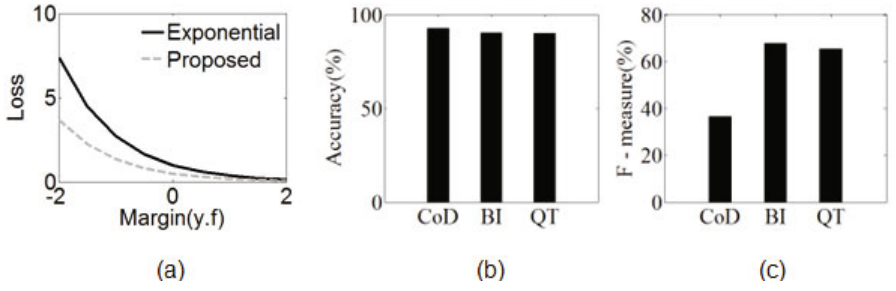


Fig. 1. (a) Loss functions for two class classification. (b) Accuracy and (c) F-measure for three different snake initialization methods.

these two methods, our method can adaptively adjust the effects of regularization in the boosting framework by selecting the proper value of k from the training data set. The regularization strategy in ϵ - boosting is imposed through shrinking the contribution of each feature (feature weight).In l_1- regularized boosting, the exponential loss function is minimized with l_1- regularization. This provides sparse solution and acts as a regularizer.

4 Results and Discussions

We have carried out experiments on two real data sets: oil sand images and leukocyte microscopy images.

4.1 Oil Sand Images

In the oil sand extraction process, oil sand ore is crushed, broken into smaller particles through crusher and then passed through screens to reject oversize ores. Undersize ores are transported to hydrotransport plant for further processing. Here, ore size is an important measure to estimate crusher as well as screen efficiency. Towards achieving this goal, oil sand images are captured through camera mounted over conveyor belt before and after the crusher as well as screen. Oil sand particles are detected in the images using the proposed method and then the particle size distribution (PSD) is computed. PSD is a histogram showing frequency of the particles over their sizes. In this paper, we have concentrated on the automatic detection of the oil sand particles. We construct a training set using 20 images and test set using 100 images sampled randomly from online video. For QT based snake initialization, we find the distribution for prior and likelihood as well as the two threshold values ((P_{th1}) and (P_{th2})) of the posterior probability ($P(O/T, B)$) experimentally from the training set. We have $P(O/T, B) \propto P(T/O)P(B/O)P(O)$, where T and B represent texture and brightness respectively. Maximum Singular Value Decomposition (SVD) encodes average brightness of a region where average of the response of the gabor filter on a region encodes texture of the region. Experimentally it is found

that maximum SVD of the oil sand patch follows doubly truncated exponential (DTE) distribution. Probability density function (pdf) of DTE [11] is given by, $P(B) = \frac{exp(-(B-\mu)/\sigma)}{\sigma[1-exp(-(x_0-\mu)/\sigma)]} I_{[\mu,x_0]}(B), \mu \leq B \leq x_0$. On the other hand, the response of the Gabor filter follows doubly truncated normal distribution (DTN); pdf of DTN is given by, $P(T) = \frac{\frac{1}{\sigma\sqrt{2\pi}} exp(-(T-\mu)^2/2\sigma^2)}{\Phi(\frac{b-\mu}{\sigma})-\Phi(\frac{a-\mu}{\sigma})} I_{[a,b]}(T), a \leq T \leq b$, where Φ is the standard normal cumulative density function(cdf) [11]. The value of indicator function, $I_{[a,b]} = 1$ if $a \leq T \leq b$, and is 0 otherwise. $I_{[\mu,x_0]}(B)$ is defined similarly. A region will have high oil sand particle density if $P(O/T, B) \geq P_{th2}$. The two threshold values of the posterior probability (P_{th1} andP_{th2}) are determined experimentally from the training set. The parameters of the above distributions are estimated using maximum likelihood estimation (MLE). Fig. 2(a) and Fig. 2(b) show the distribution of the brightness and texture of the oil sand particles respectively.

Regions of Interest (ROI) generated by QT and seeds generated by Center of Divergence (CoD) [3] method are shown in Fig. 4. Table 1 illustrates the number of seeds generated by the proposed QT, CoD and blind initialization (BI) [2]. CoD refers to the local maxima of the external Gradient Vector Flow (GVF) field. The point from which the GVF vectors to all of its neighboring pixels radiate is considered as CoD. CoD is supposed to be located within the object and the snake evolved from CoD converges to the actual boundary of the object in noise-free settings. Fig. 1(b) and 1(c) show accuracy and F-measure for CoD, BI and QT techniques with proposed modified Adaboost based validation technique respectively. F-measure combines both recall and precision into a single entity [12]. Results show that though all techniques possess the same accuracy, both BI and QT achieve 30% more F-measure value than that of CoD but QT generates significantly fewer seeds (Table 1) than other competitve methods.

Next, we determine the value of k (discussed regarding feature weight in Section 3) using five-fold cross validation [5] technique. We compute misclassification errors for different values of k and the result is shown in Fig. 3(a). Standard error bars indicate the standard errors of the individual misclassification error rates

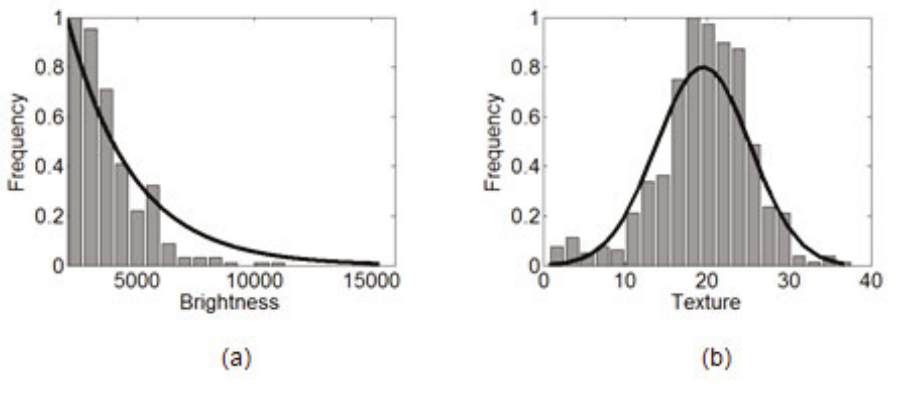


Fig. 2. Histogram of brightness and texture of oil sand particles

Table 1. Comparison among three snake initialization techniques

Datasets	# of objects	# of seeds generated by		
		CoD	BI	**QT**
Oil Sand	349	3786	3000	**686**
Leukocyte	193	2402	4375	**799**

for each of the five parts. It is observed that both the average misclassification error rate and standard error is minimum for $k = 8$ for oil sand images. For existing Adaboost algorithm, the value of k is always 1. Modified Adaboost always outperforms the existing Adaboost algorithm because the modified one can select the best value of k for which the misclassification error is minimum. The misclassification error rate for boosting with decision stumps [5], as a function of the number of iterations for $k = 8$ is shown in Fig. 3(b).

Fig. 4 shows the results of proposed Adaboost, ϵ-boosting [5], l_1 regularized boosting [10] and PCA [2] on oil sand images and their comparisons are shown in Fig. 5 and Fig. 8(a). Fig. 5(a) shows the average Jaccard Score [13] and Fig. 5(b) shows the average Pratts figure of merit (PFOM) [14] for these methods. Jaccard Score measures the fraction of overlap area among detected and true objects. Pratt's figure of merit determines the closeness among detected and actual edge pixels. Domain expert visually determines actual edge pixels and true object area from an image. Both Jaccard Score and Pratt's figure of merit are important to judge the segmentation quality of an algorithm and both are bounded by 0 and 1. Superior performance of a segmentation algorithm is indicated by higher PFOM as well as Jaccard Score values.

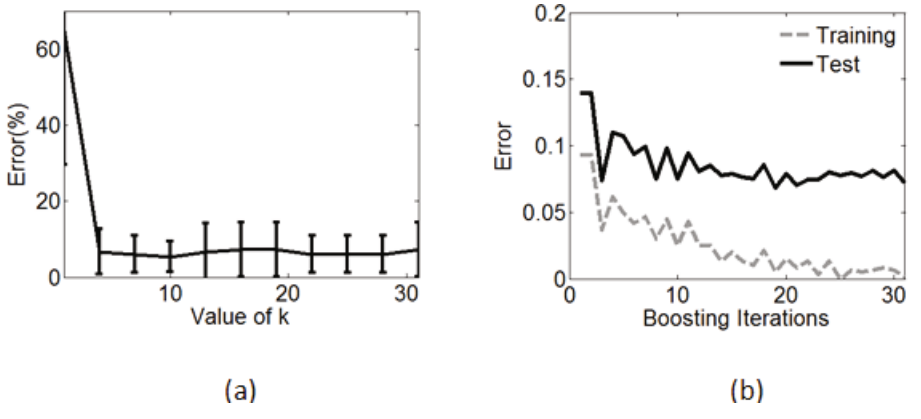


Fig. 3. (a) fivefold cross validation curve with standard error bars; the curve has minima at $k = 8$. (b) Misclassification error rate over the number of iterations for oil sand images.

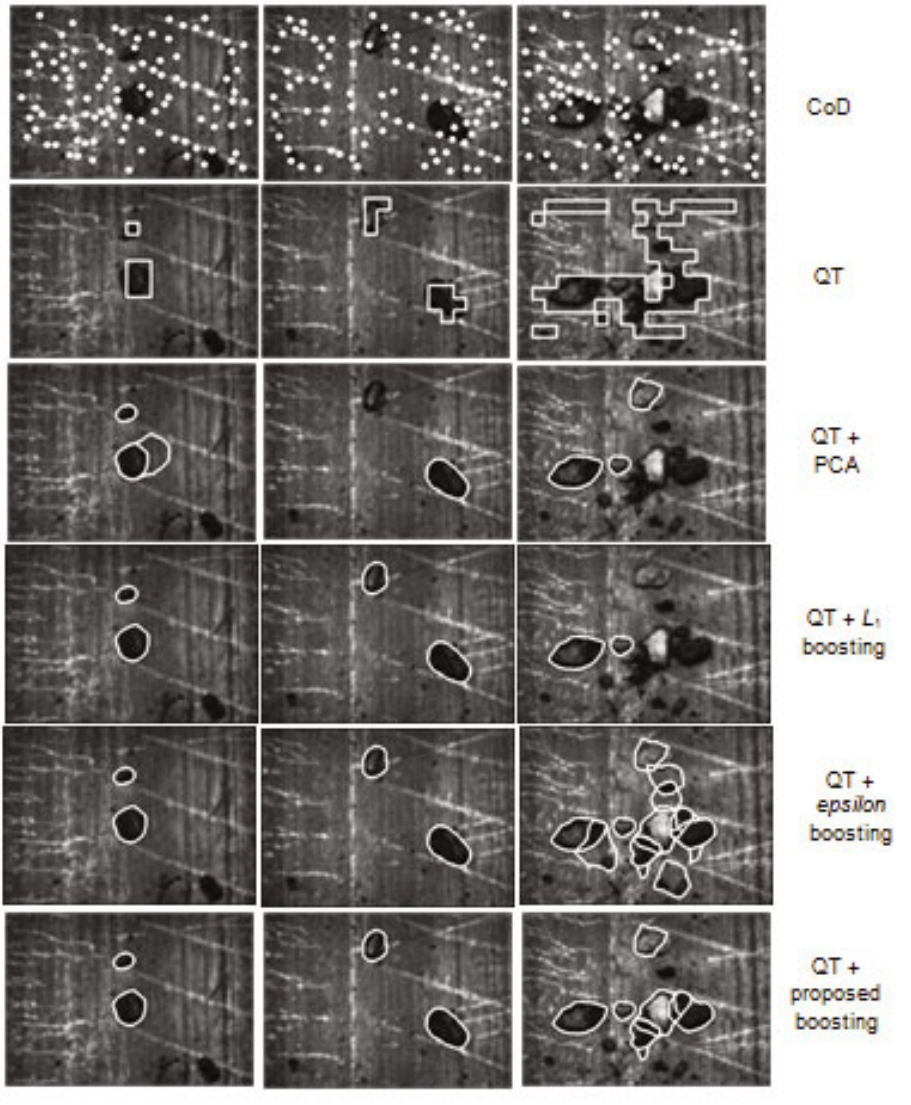


Fig. 4. Results of different methods on oil sand images

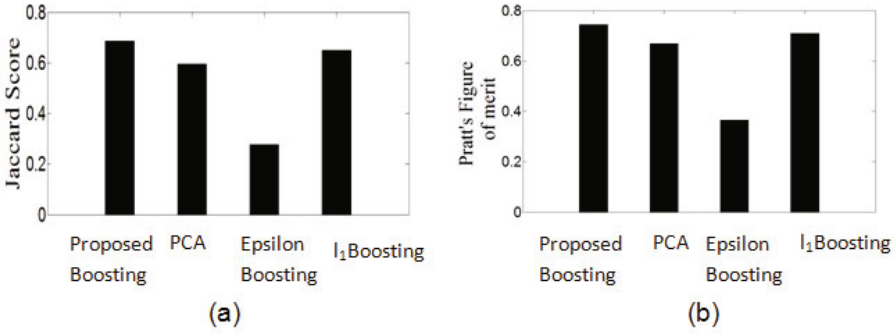


Fig. 5. Segmentation scores: (a) Jaccard Score and (b) Pratt's Figure of Merit of different methods on oil sand images

4.2 Leukocyte Images

Leukocyte plays an important role in the study of inflammation. Inflammation is a natural defense mechanism initiated by tissue damage. During inflammatory responses, endothelium cells are activated, and then leukocytes start deviating from mainstream blood flow and contact the activated endothelium cells. This slow movement of leukocytes in contact with endothelium cells is known as rolling. Finally, from the rolling stage, leukocyte diffuses through the vascular wall, reaches the injured tissues, and encounters the germs. Although inflammation is a normal defense mechanism, it sometimes becomes an abnormality in the context of inflammatory diseases. To combat such diseases, anti-inflammatory drugs are developed by blocking or controlling any of the necessary processes of inflammatory response. Here, the rolling velocity distributions of leukocytes is an important factor in the study of inflammation. To measure and analyze the rolling velocity distributions of leukocytes from the *in vivo* experiments, video recordings of the postcapilary vennule of a cremaster muscle are made through a CCD camera coupled with the intravital microscope. Then leukocytes are detected from the video frames using the proposed method and a correspondence analysis is carried out between consecutive images to compute their velocities [6]. In this paper, we have concentrated only leukocyte detection. We have carried out experiment on a training set of 5 and a test set of 25 leukocyte images. Detections obtained by proposed Adaboost, ϵ-boosting [5], l_1 regularized boosting [10] and PCA [2] techniques are shown in Fig. 6 and their performances in terms of Jaccard Score and Pratt's Figure of Merit are shown in Fig. 7 and Fig. 8(b).

4.3 Interpretation of Results

One can interpret that proposed adaboost based validation is better than ϵ-boosting [5], l_1 regularized boosting [10] and PCA [2] based technique since it can detect more oil sand particles and leukocytes accurately and precisely.

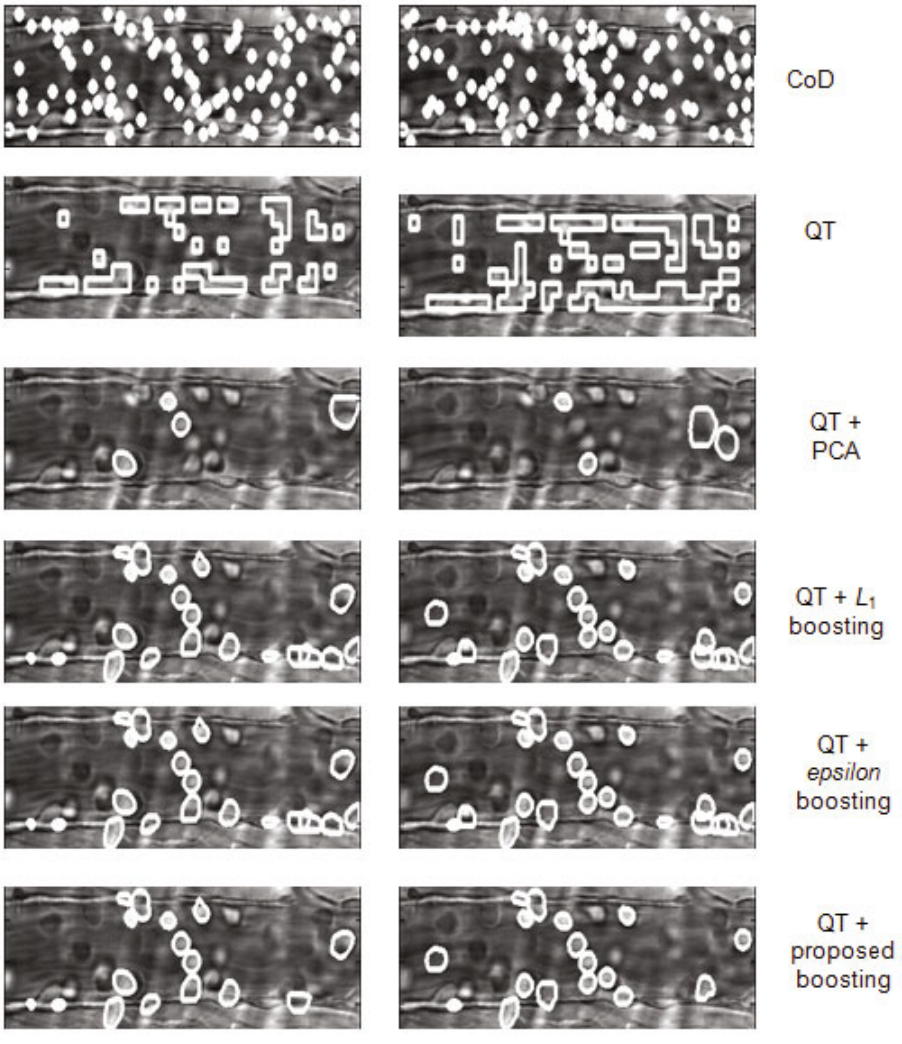


Fig. 6. Results of different techniques on leukocyte images

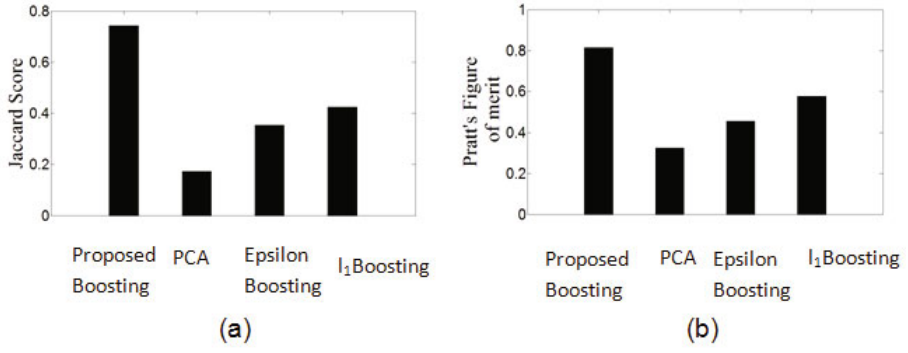


Fig. 7. Segmentation scores: (a) Jaccard Score and (b) Pratt's Figure of Merit of different methods on leukocyte images

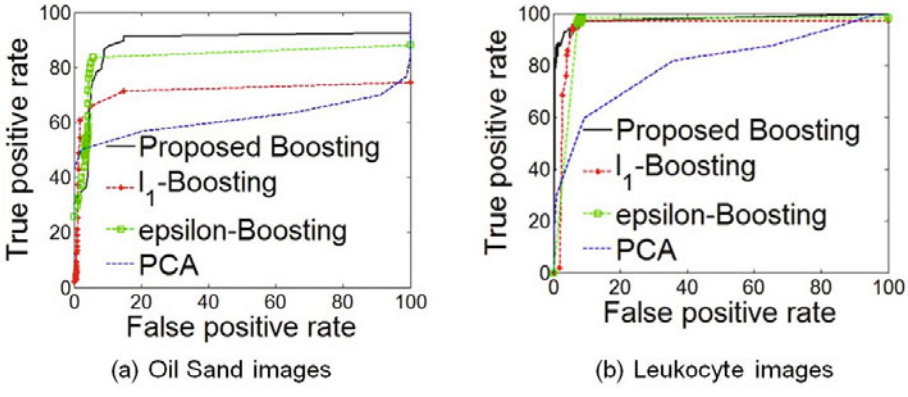


Fig. 8. Receiver Operating Characteristic (ROC) curves

Segmentation score (Jaccard Score and Pratt's Figure of Merit) as well as area under ROC curve of proposed adaboost is greater than that of other methods.

5 Conclusion and Fututre Works

Towards complete automation of snake algorithm, we have proposed an initialization as well as validation algorithm that could be utilized as a successful plug-in for existing snake/active contour tools. Existing research mainly focuses on the snake initialization and evolution steps and ignores the validation step. Here, we emphasize that we cannot omit the validation step in spite of applying the smart initialization technique of snake algorithm used for multiple objects detection. We have proposed probabilistic quad tree based approximate segmentation for snake initialization. We show that our proposed initialization

outperforms existing initialization methods. We have successfully incorporated regularization into boosting framework and we demonstrate that our intended loss function is more robust to outliers concerning snake classification into object and non-object classes. We have also shown that proposed boosting based snanke validation technique outperforms existing PCA based validation method. Results of extensive experiments illustrate that proposed method is fast, reliable and more accurate than existing methods.

In the future, We would like to incorporate our initialization and validation methods with other well-known snake evolution methods. Also we will further explore the characteristics of proposed regularization into boosting frameworks extensively by conducting experiments with available benchmark datasets.

Acknowledgements. The authors acknowledge the support of NSERC, Department of Computing Science, University of Alberta, and the Center for Intelligent Mining Systems (CIMS), University of Alberta, Mitacs internship program for this work.

References

1. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: active contour models. IJCV 1, 321–331 (1987)
2. Saha, B.N., Ray, N., Zhang, H.: Snake validation: A pca-based outlier detection method. IEEE Signal Processing Letters 16, 549–552 (2009)
3. Ge, X., Tian, J.: An automatic active contour model for multiple objects. In: ICPR, vol. 2, pp. 881–884 (2002)
4. Saha, B.N., Ray, N., Zhang, H.: Computing oil sand particle size distribution by snake-pca algorithm. In: ICASSP, pp. 977–980 (2008)
5. Hastie, T., Tibshirani, R., Friedman, J.: The elements of statistical learning: Data mining, inference, and prediction, 2nd edn. Springer, Heidelberg (2009)
6. Dong, G., Ray, N., Acton, S.T.: Intravital leukocyte detection using the gradient inverse coefficient of variation. IEEE Transaction on Medical Imaging 24, 910–924 (2005)
7. Mirmehdi, M., Xie, X., Suri, J.: Handbook of texture analysis. Imperial college Press, London (2008)
8. Omerevi, D., Perko, R., Targhi, A.T., Eklundh, J.O., Leonardis, A.: Vegetation segmentation for boosting performance of mser feature detector. In: Computer Vision Winter Workshop, pp. 17–23 (2008)
9. Russ, J.C.: The image processing handbook, 3rd edn. CRC & IEEE press (1995)
10. Xi, Y.T., Xiang, Z.J., Ramadge, P.J., Schapire, R.E.: Speed and sparsity of regularized boosting. In: 12th International Conference on Artificial Intelligence and Statistics (AISTATS), vol. 5, pp. 615–622 (2009)
11. Maritz, J.S.: Distribution-free statistical methods, 2nd edn. Second Edition. Chapman & Hall, Boca Raton (1995)
12. van Rijsbergen, C.J.: Information retireval. Butterworths, London (1979)
13. Jaccard, P.: Distribution de la flore alpine dans le bassin des dranses et dans quelques rgions voisines. Bulletin de la Socit Vaudoise des Sciences Naturelles 37, 241–272 (1901)
14. Abdou, I.E., Pratt, W.K.: Quantitative design and evaluation of enhancement/ thresholding edge detectors. Proceedings of the IEEE 67, 753–763 (1979)

Appendix: Derivation of Proposed Discrete Adaboost Algorithm

Proposed loss function is: $L(y, f(x)) = exp(-yf(x) + \lambda|y - G(x)|)$, where $\lambda < 0$. Let $f_m(x) = f_{m-1}(x) + \beta_m G_m(x)$ be the strong classifier composed of first m classifiers. We can pose m-th iteration of adaboost as the following optimization, $(\beta_m, G_m) = \underset{\beta,G}{argmin} \sum_{i=1}^{N} exp[-y_i(f_{m-1}(x_i) + \beta G(x_i)) + \lambda|y_i - G(x_i)|]$

$\Rightarrow (\beta_m, G_m) = \underset{\beta,G}{argmin} \sum_{i=1}^{N} w_i^m exp[-y_i \beta G(x_i)) + \lambda|y_i - G(x_i)|]$

where, $w_i^m = exp(-y_i f_{m-1}(x_i))$ is free of both β and $G(x)$.

$\Rightarrow (\beta_m, G_m) = \underset{\beta,G}{argmin}[exp(-\beta) \sum_{i:y_i=G(x_i)} w_i^m + exp(\beta + 2\lambda) \sum_{i:y_i \neq G(x_i)} w_i^m]$.

$= \underset{\beta,G}{argmin}[exp(\beta + 2\lambda) - exp(-\beta)) \sum_{i:y_i \neq G(x_i)} w_i^m + exp(-\beta) \sum_{i=1}^{N} w_i^m]$.

The solution forβ_m and G_m can be obtained in two steps. First, for any value of $\beta > 0$, the solution for G_m is: $G_m = \underset{G}{argmin} \sum_{i=1}^{N} w_i^m I(y_i \neq G(x_i))$.

Let $err_m = \underset{G}{argmin} \sum_{i=1}^{N} w_i^m I(y_i \neq G(x_i)) / \sum_{i=1}^{N} w_i^m$,

then $\beta_m = \frac{\partial}{\partial \beta}(\sum_{i=1}^{N} w_i^m((exp(\beta + 2\lambda) - exp(-\beta))err_m + exp(-\beta))) = 0$.

$\Rightarrow \beta_m = \frac{1}{2}(log\frac{1-err_m}{err_m}) - \lambda = \frac{1}{2}(logk\frac{1-err_m}{err_m})$, where, $\lambda = -\frac{1}{2}log(k), k > 0$.

Now, $w_i^{m+1} = w_i^m exp(-\beta_m y_i G_m(x_i))$. Using the fact that $-y_i G_m(x_i) = 2I(y_i \neq G(x_i)) - 1$, we get, $w_i^{m+1} = w_i^m exp(\alpha_m I(y_i \neq G(x_i)))exp(-\beta_m)$

where, $\alpha_m = 2\beta_m = log(k((1 - err_m)/err_m))$. So, $w_i^{m+1} = w_i^m exp(\alpha_m I(y_i \neq G(x_i)))$. The factor $exp(-\beta_m)$ multiplies all weights by the same value, so it has no effect.

Monocular Template-Based Reconstruction of Smooth and Inextensible Surfaces

Florent Brunet[1,2], Richard Hartley[3], Adrien Bartoli[1],
Nassir Navab[2], and Remy Malgouyres[4]

[1] ISIT, Université d'Auvergne, Clermont-Ferrand, France
[2] CAMPAR, Technische Universtät München, Germany
[3] Research School of Information Sciences, ANU, NICTA, Australia
[4] LIMOS, UMR 6158, Clermont-Ferrand, France

Abstract. We present different approaches to reconstructing an inextensible surface from point correspondences between an input image and a template image representing a flat reference shape from a fronto-parallel point of view. We first propose a 'point-wise' method, *i.e.* a method that only retrieves the 3D positions of the point correspondences. This method is formulated as a second-order cone program and it handles inaccuracies in the point measurements. It relies on the fact that the Euclidean distance between two 3D points must be shorter than their geodesic distance (which can easily be computed from the template image). We then present an approach that reconstructs a smooth 3D surface based on Free-Form Deformations. The surface is represented as a smooth map from the template image space to the 3D space. Our idea is to say that the 2D-3D map must be everywhere a local isometry. This induces conditions on the Jacobian matrix of the map which are included in a least-squares minimization problem.

1 Introduction

Monocular surface reconstruction of deformable objects is a challenging problem which has known renewed interest during the past few years. This problem is fundamentally ill-posed because of the depth ambiguities; there are virtually an infinite number of 3D surfaces that have exactly the same projection. It is thus necessary to use additional constraints ensuring the consistency of the reconstructed surface.

In this paper, we present two algorithms for monocular reconstruction of deformable and inextensible surfaces under some general assumptions. First, we consider the *template-based* case. Reconstruction is achieved from point correspondences between an input image and a template image showing a flat reference shape from a fronto-parallel point of view. Second, we suppose the intrinsic parameters of the camera to be known. Third, we assume that the camera is a perspective camera. These are common assumptions [1–3].

Over the years, different types of constraints have been proposed to disambiguate the problem of monocular reconstruction of deformable surfaces.

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 52–66, 2011.

They can be divided into two main categories: the *statistical* and the *physical* constraints. For instance, the methods relying on the low-rank factorization paradigm [4–10] can be classified as statistical approaches. Learning approaches such as [1, 11–13] also belong to the statistical approaches. Work such as [1], where the reconstructed surface is represented as a linear combination of inextensible deformation modes, is also a statistical approach. Physical constraints include spatial and temporal priors on the surface to reconstruct [14, 15]. Statistical and physical priors can be combined [5, 7]. A physical prior of particular interest is the hypothesis of having an inextensible surface [1–3, 16]. In this paper, we consider this type of surface. This hypothesis means that the geodesics on the surface may not change their length across time. However, computing geodesics is generally hard to achieve and it is even more difficult to incorporate such constraints in a reconstruction algorithm. There exist several approaches to approximate this type of constraint. For instance, if the points are sufficiently close together, the geodesic between two 3D points on the surface can be approximated by the Euclidean distance [17]. An efficient approximation consists in saying that the geodesic distance between two points is an upper bound to the Euclidean distance [3, 16].

Algorithms for monocular reconstruction of deformable surfaces can also be categorized according to the type of surface model (or representation) they use. The *point-wise* methods utilize a sparse representation of the 3D surface, *i.e.* they only retrieve the 3D positions of the data points [3]. Other methods use more complex surface models such as triangular meshes [1, 16] or smooth surfaces such as Thin-Plate Splines [3, 5]. In this latter case, the 3D surface is represented as a parametric 2D-3D map between the template image space and the 3D space. Smooth surfaces are generally obtained by fitting a parametric model to a sparse set of reconstructed 3D points: the smooth surface is not actually used in the 3D reconstruction process. In this paper, we propose an algorithm that directly estimates a smooth 3D surface based on Free-Form Deformations [18]. Having an inextensible surface means that the surface must be everywhere a local isometry. This induces conditions on the Jacobian matrix of the 2D-3D map. We show that these conditions can be integrated in a non-linear least-squares minimization problem along with some other constraints that force the consistency between the reconstructed surface and the point correspondences. Such a problem can be solved using an iterative optimization procedure such as Levenberg-Marquardt that we initialize using a point-wise reconstruction algorithm. Our approach is highly effective in the sense that it outperforms previous approaches in terms of accuracy of the reconstructed surface and in terms of inextensibility.

Another important aspect in monocular reconstruction of deformable surfaces is the way noise is handled. It can be accounted for in the template image [3] or in the input image [1]. There exist different approaches for handling the noise. For instance, one can minimize a reprojection error, *i.e.* the distance between the data points of the input image and the projection of the reconstructed 3D points. It is also possible to hypothesize maximal inaccuracies in the data points.

Table 1. Notation used in this paper

Notation	Description
P	Matrix of the intrinsic parameters of the camera ($\mathsf{P} \in \mathbb{R}^{3\times3}$) (The camera is assumed to be at the coordinate origin, so the matrix P may be assumed to be square and invertible.)
$\mathbf{p}_k^\mathsf{T}$	kth row of the matrix P
n_c	Number of point correspondences
$\mathbf{q}_i$	ith point in the template image
$\mathbf{q}'_i$	ith point in the input image; $i \in \{1, \ldots, n_c\}$
$\bar{\mathbf{q}}_i$	Point $\mathbf{q}_i$ in homogeneous coordinates
$\mathbf{u}_i$	Sightline corresponding to the point $\mathbf{q}'_i$ ($\mathbf{u}_i = (\mathsf{P}^{-1}\bar{\mathbf{q}}'_i)/\|\mathsf{P}^{-1}\bar{\mathbf{q}}'_i\|$)
μ_i	Depth of the point $\mathbf{Q}_i$
$\mathbf{Q}_i$	Reconstructed 3D point i
d_{ij}	Euclidean distance between points i and j ($d_{ij} = \|\mathbf{q}_i - \mathbf{q}_j\|$)
$\hat{x}$	True value of x (for $x = \mathbf{q}'_i, \mathbf{q}_i, \mathbf{Q}_i, \mathbf{u}_i, \mu_i, d_{ij}$)

We propose a point-wise approach that accounts for noise in both the template and the input images. This approach is formulated as a second-order cone program (SOCP) [19].

2 Related Work on Inextensible Surface Reconstruction

A popular assumption made in deformable surface reconstruction is to consider that the surface to reconstruct is inextensible [1–3, 16]. This assumption is reasonable for many types of material such as paper and some types of fabrics. Having an inextensible surface means that the surface is an isometric deformation of the reference shape. Another way of putting it is to say that the length of the geodesics between pairs of points remains unchanged when the surface deforms. An exact transcription of this principle is difficult to integrate in a reconstruction algorithm. Indeed, while it is trivial to compute the geodesic in a flat reference shape, it is quite difficult to do it for a bent surface (especially when the surface is represented as a sparse set of points or a triangular mesh). Many approximations have thus been proposed.

The first type of approximation consists in saying that if the surface does not deform too much then the Euclidean distance is a good approximation to the geodesic distance. Such an approach has been used for instance in [2, 12, 16, 20]. Note that these types of constraints are usually set in a soft way. For a given set of point pairs on the surface, the Euclidean distance should not diverge too much from the geodesic distances. This approximation is better when there are a large number of points. Depending on the surface model it is not always possible to vary the number of points.

Although the Euclidean approximation can work well in some cases, this approximation gives poor results when creases appear in the 3D surface. In this

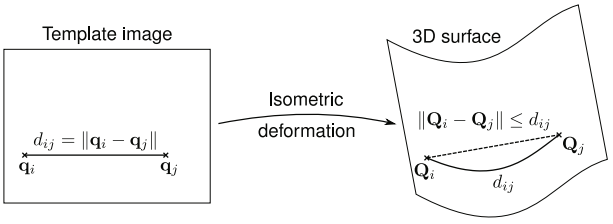


Fig. 1. Inextensible object deformation. The Euclidean distance between two points is necessarily less than or equal to the length of the geodesic that links those two points (this length is easily computable if we have a template image representing the flat reference surface from a fronto-parallel point of view).

case, the Euclidean distance between two points on the surface can shrink, as illustrated in figure 1. The 'upper bound approach' is a now classical approach [1, 3] which consists in noticing that even if the Euclidean distance between two points can shrink it can never be greater than the length of the corresponding geodesic. In other words, the *inextensibility constraint* $\|\mathbf{Q}_i - \mathbf{Q}_j\| \leq d_{ij}$ must be satisfied for any pair of points $(\mathbf{Q}_i, \mathbf{Q}_j)$ lying on the surface. The second principle of such algorithms is to say that a 3D point $\mathbf{Q}_i$ must lie on the sight-line $\mathbf{u}_i$, *i.e.* $\mathbf{Q}_i = \mu_i \mathbf{u}_i$. These two constraints are not sufficient to reconstruct the surface. Indeed, nothing prevents the reconstructed surface from shrinking towards the optical centre of the camera. This problem is 'solved' using a heuristic that has been proven to be very effective in practice. It consists in considering a perspective camera and in maximizing the depth of the reconstructed 3D points.

These ideas have been implemented in different manners. For instance, [3] proposes a dedicated algorithm that enforces the inextensibility constraints. This algorithm accounts for noise only in the template image (by simply increasing a little bit the geodesic distances in the template, *i.e.* by replacing d_{ij} with $d_{ij} + \varepsilon_{\mathcal{T}}$ where $\varepsilon_{\mathcal{T}}$ is the maximal inaccuracy of the points in the template image). Another sort of implementation is given by [1, 16]. In these papers, a convex cost function combining the depth of the reconstructed points and the negative of the reprojection error is maximized while enforcing the inequality constraints arising from the surface inextensibility. The resulting formulation can be easily turned into an SOCP problem. A similar approach is explored in [2]. These last two methods account for noise in the input image. The approach of [3] is a point-wise method. The approaches of [1, 2, 16] use a triangular mesh as surface model, and the inextensibility constraints are applied to the vertices of the mesh.

3 Convex Formulation of the Upper Bound Approach with Noise in All Images

In this section, we propose a convex formulation of the principles sketched in §2 that, compared to [3], accounts for noise in both the template and the input images. We can express this in terms of image-plane measurements. As in [1, 16],

our approach is formulated as an SOCP problem. However, contrary to [1, 16], our approach is a point-wise method that does not require us to tune the relative influence of minimizing the reprojection error and maximizing the depths.

3.1 Noise in the Template Only

Let us first remark that the basic principles explained in §2 can be formulated as SOCP problems. In this first formulation, the noise is only account for in the template image. The inextensibility constraint $\|\mathbf{Q}_i - \mathbf{Q}_j\| \leq d_{ij} + \varepsilon_{\mathcal{I}}$ can be written:

$$\|\mu_i \mathbf{u}_i - \mu_j \mathbf{u}_j\| \leq d_{ij} + \varepsilon_{\mathcal{I}}. \tag{1}$$

Including the maximization of the depths, we obtain this SOCP problem:

$$\begin{aligned} \max_{\boldsymbol{\mu}} \ & \sum_{i=1}^{n_c} \mu_i \\ \text{subject to } & \|\mu_i \mathbf{u}_i - \mu_j \mathbf{u}_j\| \leq d_{ij} + \varepsilon_{\mathcal{I}} \quad \forall (i,j) \in \mathcal{E} \\ & \mu_i \geq 0 \qquad\qquad\qquad\qquad i \in \{1, \ldots, n_c\} \end{aligned} \tag{2}$$

where $\boldsymbol{\mu}^\mathsf{T} = \begin{pmatrix} \mu_1 & \ldots & \mu_{n_c} \end{pmatrix}$, and $\mathcal{E}$ is a set of pairs of points to which the inextensibility constraints are applied.

3.2 Noise in Both the Template and the Input Images

Let us now suppose that the inaccuracies are expressed in terms of image-plane measurements. Suppose that points are measured in the image with a maximum error of $\varepsilon_{\mathcal{I}}$, *i.e.*

$$\|\hat{\mathbf{q}}'_i - \mathbf{q}'_i\| \leq \varepsilon_{\mathcal{I}}, \qquad \forall i \in \{1, \ldots, n_c\}. \tag{3}$$

Since we are searching for the true 3D position of the point $\mathbf{Q}_i$, we say that:

$$\hat{\mathbf{q}}'_i = \frac{1}{\mathbf{p}_3^\mathsf{T} \mathbf{Q}_i} \begin{pmatrix} \mathbf{p}_1^\mathsf{T} \mathbf{Q}_i \\ \mathbf{p}_2^\mathsf{T} \mathbf{Q}_i \end{pmatrix}. \tag{4}$$

Equation (3) can thus be rewritten:

$$\left\| \frac{1}{\mathbf{p}_3^\mathsf{T} \mathbf{Q}_i} \begin{pmatrix} \mathbf{p}_1^\mathsf{T} \mathbf{Q}_i \\ \mathbf{p}_2^\mathsf{T} \mathbf{Q}_i \end{pmatrix} - \mathbf{q}'_i \right\| \leq \varepsilon_{\mathcal{I}}. \tag{5}$$

We finally add the inextensibility constraints and the maximization of the depths (which are given by $\mathbf{p}_3^\mathsf{T} \mathbf{Q}_i$) and we obtain the following SOCP problem:

$$\begin{aligned} \max_{\mathbf{Q}} \ & \mathbf{p}_3^\mathsf{T} \sum_{i=1}^{n_c} \mathbf{Q}_i \\ \text{subject to } & \left\| \begin{bmatrix} \mathbf{p}_1^\mathsf{T} \\ \mathbf{p}_2^\mathsf{T} \end{bmatrix} \mathbf{Q}_i - \mathbf{q}'_i \mathbf{p}_3^\mathsf{T} \mathbf{Q}_i \right\| \leq \varepsilon_{\mathcal{I}} \, \mathbf{p}_3^\mathsf{T} \mathbf{Q}_i \quad \forall i \in \{1, \ldots, n_c\} \\ & \|\mathbf{Q}_i - \mathbf{Q}_j\| \leq d_{ij} \qquad\qquad \forall (i,j) \in \mathcal{E} \\ & \mathbf{p}_3^\mathsf{T} \mathbf{Q}_i \geq 0 \qquad\qquad\qquad \forall i \in \{1, \ldots, n_c\} \end{aligned} \tag{6}$$

where $\mathbf{Q}$ is the concatenation of the 3D points $\mathbf{Q}_i$, for $i \in \{1, \ldots, n_c\}$.

4 Smooth and Inextensible Surface Reconstruction

Although the strategem of maximizing the sum of depths $\sum_{i=1}^{n_c} \mu_i$ described in the previous section gives reasonable results, it is merely a heuristic, not based on any valid principle related to surface properties. We therefore consider next a new formulation based on the principle of surface inextensibility.

Let the surface be modelled as a function $\mathcal{W} : \mathbb{R}^2 \to \mathbb{R}^3$, mapping the planar template to 3-dimensional space. The inextensibility constraint is equivalent to saying that the map $\mathcal{W}$ must be everywhere a local isometry. This condition may be expressed in terms of its Jacobian. Let $\mathsf{J}(\mathbf{q}) \in \mathbb{R}^{3\times 2}$ be the Jacobian matrix $\partial\mathcal{W}/\partial\mathbf{q}$ evaluated at the point $\mathbf{q}$. The map $\mathcal{W}$ is an isometry at $\mathbf{q}$ if the columns of $\mathsf{J}(\mathbf{q})$ are orthonormal. This local isometry can be enforced for the whole surface with the following least-squares constraint:

$$\iint \left\| \mathsf{J}(\mathbf{q})^\mathsf{T} \mathsf{J}(\mathbf{q}) - \mathsf{I}_2 \right\|^2 \mathrm{d}\mathbf{q} = 0. \tag{7}$$

In practice, we consider a discretization of the quantity in equation (7), namely

$$\mathcal{E}_i(\mathcal{W}) = \sum_{j=1}^{n_j} \left\| \mathsf{J}(\mathbf{g}_j)^\mathsf{T} \mathsf{J}(\mathbf{g}_j) - \mathsf{I}_2 \right\|^2, \tag{8}$$

where $\{\mathbf{g}_j\}_{j=1}^{n_j}$ is a set of 2D points in the template image space taken on a fine and regular grid (for instance, a grid of size 30×30). This term $\mathcal{E}_i(\mathcal{W})$ measures the departure from inextensibility of the surface $\mathcal{W}$.

Our minimization problem is then to minimize this quantity, over all possible surfaces, subject to the projection constraints, namely that point $\mathcal{W}(\mathbf{q}_i)$ projects to (or near to) the image point $\mathbf{q}'_i$, for all i.

4.1 Parametric Surface Model

The problem just described involves a minimization over all possible surfaces. Instead of considering this as a variational problem over all possible surfaces, we consider a parametrized family of surfaces. For this purpose, we chose Free-Form Deformations (FFD) [18] based on uniform cubic B-splines [21]. Let $\mathcal{W}_{\boldsymbol{\ell}} : \mathbb{R}^2 \to \mathbb{R}^3$ be the parametric FFD, parametrized by a family of 3D points $\boldsymbol{\ell}_{jk}$; $j \in \{1, \ldots, n_u\}$, $k \in \{1, \ldots, n_v\}$, which act as 'attractors' for the surface.

For a point $\mathbf{q} = (u, v)$ in the template, the surface point is explicitly given as

$$\mathcal{W}_{\boldsymbol{\ell}}(\mathbf{q}) = \sum_{j=1}^{n_u} \sum_{k=1}^{n_v} \boldsymbol{\ell}_{jk} N_j(u) N_k(v). \tag{9}$$

The functions N_j are the B-spline basis functions [21] which are polynomials of degree 3. If point $\mathbf{q}_i = (u_i, v_i)$ is fixed and known then the surface point $\mathcal{W}_{\boldsymbol{\ell}}(\mathbf{q}_i)$ is expressed as a linear combination of the points $\boldsymbol{\ell}_{jk}$, and hence can be written in the form $\mathcal{W}_{\boldsymbol{\ell}}(\mathbf{q}_i) = \mathsf{W}_i \boldsymbol{\ell}$, where W_i is a $3 \times n_u n_v$ matrix depending only on the

point $\mathbf{q}_i$, and $\boldsymbol{\ell}$ is the vector obtained by concatenating all the points $\boldsymbol{\ell}_{jk}$. Thus, the 3D point is a linear expression in terms of the parameter vector $\boldsymbol{\ell}$. Since the polynomials N_j and N_k depend only on a local set of the attractor points $\boldsymbol{\ell}_{jk}$, the matrix W_i is sparse, which is important for computational efficiency.

4.2 Surface Reconstruction as a Least-Squares Problem

By replacing $\mathbf{Q}_i$ by $\mathsf{W}_i\boldsymbol{\ell}$ in (6) we may arrive at a constraint:

$$\left\| \left(\begin{bmatrix} \mathbf{p}_1^\mathsf{T} \\ \mathbf{p}_2^\mathsf{T} \end{bmatrix} - \mathbf{q}_i' \mathbf{p}_3^\mathsf{T} \right) \mathsf{W}_i \boldsymbol{\ell} \right\| \leq \varepsilon_\mathcal{I} \mathbf{p}_3^\mathsf{T} \mathsf{W}_i \boldsymbol{\ell}. \tag{10}$$

We may then formulate the optimization problem as minimizing the inextensibility cost $\mathcal{E}_i(\mathcal{W}_{\boldsymbol{\ell}})$ given in (8) over all choices of parameters $\boldsymbol{\ell}$, subject to constraints (10). The constraints are SOCP constraints, but the cost function (8) is of higher degree in the parameters. To avoid the difficulties of constrained non-linear optimization, we choose a different course, by including the reprojection error into the cost function, leading to an unconstrained problem.

To simplify the formulation of the reprojection error, we introduce the depths μ_i as subsidiary variables, for reasons that become evident below. This is not strictly necessary, but reduces the degree of the reprojection-error term. The minimization problem now takes the form:

$$\min_{\boldsymbol{\mu},\boldsymbol{\ell}} \mathcal{E}_d(\boldsymbol{\mu},\boldsymbol{\ell}) + \alpha\mathcal{E}_i(\boldsymbol{\ell}) + \beta\mathcal{E}_s(\boldsymbol{\ell}), \tag{11}$$

where $\mathcal{E}_d$, $\mathcal{E}_i$, $\mathcal{E}_s$ are the *data* (reprojection error), *inextensibilty*, and *smoothing* terms respectively. The data term ensures the consistency of the point correspondences with the reconstructed surface. $\mathcal{E}_i$ forces the inextensibility of the surface. $\mathcal{E}_s$ promotes smooth surface in order to cope with, for instance, lack of data. The relative influence of these three terms are controlled with the weights $\alpha \in \mathbb{R}_+$ and $\beta \in \mathbb{R}_+$.

The inextensibility term has been described previously. We now describe the two other terms in (11).

Data term. Replacing $\mathbf{Q}_i$ by $\mathsf{W}_i\boldsymbol{\ell}$ in (5) gives an expression for the reprojection error associated with some point. However, the resulting expression is non-linear with respect to the parameters $\boldsymbol{\ell}$. We thus prefer a linear data term expressed in terms of '3D errors', which is the reason why we introduced the depths $\boldsymbol{\mu}$ of the data points in the optimization problem. The data term is then defined by:

$$\mathcal{E}_d(\boldsymbol{\mu},\boldsymbol{\ell}) = \sum_{i=1}^{n_c} \left\| \mathcal{W}_{\boldsymbol{\ell}}(\mathbf{q}_i) - \mu_i \mathsf{P}^{-1} \bar{\mathbf{q}}_i' \right\|^2, \tag{12}$$

which measures the distance between the point $\mathcal{W}_{\boldsymbol{\ell}}$ on the surface and the point at depth μ_i along the ray defined by $\mathbf{q}_i'$.

Smoothing term. In some cases, the point correspondences and the hypothesis of an inextensible surface are not sufficient. For instance, imagine that there is no point correspondence in a corner of the surface. In this case, there is nothing that indicates how the surface should behave. The corners of the surface can bend freely as long as they do not extend or shrink (like the corners of a piece of paper). To overcome this difficulty, we can add a third term (the smoothing term) in our cost function that favours non-bending surfaces. Note that usually, such terms are used to compensate for the undesirable effects of under-fitting and over-fitting. Doing so is usually a problem because it requires one to determine a correct value for the weight associated to the smoothing term (value β in equation (11)). This is a sensible and critical way of balancing the effective complexity of the surface against the complexity of the data. Here, we do not have to care too much. Indeed, the complexity of the surface is limited by the fact that it is inextensible. Any small value (but big enough to be not negligible, for instance $\beta = 10^{-4}$) is thus suitable for the weight of the smoothing term. We define our smoothing term using the bending energy:

$$\mathcal{E}_s(\boldsymbol{\mu}, \boldsymbol{\ell}) = \sum_{i=1}^{3} \iint \left\| \frac{\partial^2 \mathcal{W}^i_{\boldsymbol{\ell}}(\mathbf{q})}{\partial \mathbf{q}^2} \right\|^2_{\mathcal{F}} \mathrm{d}\mathbf{q}. \tag{13}$$

where $\mathcal{W}^i_{\boldsymbol{\ell}}(\mathbf{q})$ is the i-th coordinate of the point, and $\|\cdot\|_{\mathcal{F}}$ is the Frobenius norm of the Hessian matrix. With FFD, there exists a simple and linear closed-form expression for the bending energy:

$$\mathcal{E}_s(\boldsymbol{\ell}) = \|\mathsf{B}^{1/2}\boldsymbol{\ell}\|^2 = \boldsymbol{\ell}^{\mathsf{T}}\mathsf{B}\boldsymbol{\ell} \tag{14}$$

where $\mathsf{B} \in \mathbb{R}^{3p \times 3p}$ is a symmetric, positive, and semi-definite matrix which can be easily computed from the second derivatives of the B-spline basis functions.

Initial solution. The problem of equation (11) is a non-linear least-squares minimization problem typically solved using an iterative scheme such as Levenberg-Marquardt. Such an algorithm requires a correct initial solution. We used an FFD surface fitted to the 3D points reconstructed with one of the point-wise methods presented in §3. Subsequently, since we use a surface model which is linear with respect to its parameters, the initial parameters $\boldsymbol{\ell}$ can be found by solving the least-squares problem:

$$\min_{\boldsymbol{\ell}} \sum_{i=1}^{n_c} \left\| \mathcal{W}_{\boldsymbol{\ell}}(\mathbf{q}_i) - \mathbf{Q}_i \right\|^2 \Leftrightarrow \min_{\boldsymbol{\ell}} \sum_{i=1}^{n_c} \|\mathsf{W}_i\boldsymbol{\ell} - \mathbf{Q}_i\|^2 \ . \tag{15}$$

An alternative is to modify the problem (6), expressing $\mathbf{Q}_i$ in terms of the required parameters $\boldsymbol{\ell}$, according to $\mathbf{Q}_i = \mathsf{W}_i\boldsymbol{\ell}$. Then one may solve for $\boldsymbol{\ell}$ directly using SOCP. If necessary, the linear smoothing term of equation (13) can be included in equation (15).

5 Experimental Results

5.1 Experiments on Synthetic Data

In this section, we experiment several aspects of different reconstruction algorithms. We first use synthetic piece of papers, such as those of figure 2, randomly generated using the code provided by [22]. The piece of papers are square and 200mm wide. The input images are simulated by projecting the deformed piece of paper with a virtual camera placed at approximately 1 meter of the paper sheet and with a focal length of 36mm. A set of n_c point correspondences are generated by taking random locations on the 3D surface. A zero mean Gaussian noise with standard deviation of 1 pixel is added to the point correspondences. There are no self-occlusion in the data.

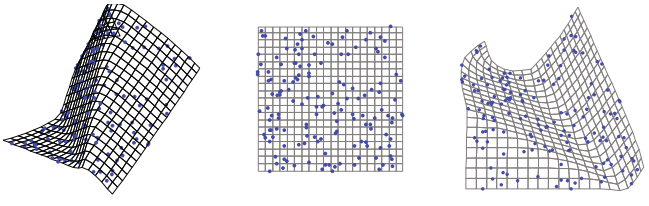


Fig. 2. Example of randomly generated piece of paper. Left: 3D surface. Middle: template image. Right: input image. The blue dots are examples of point correspondences.

Several algorithms are compared in our experiments:

- SOCPimg: our point-wise method described in §3.2 ;
- FFDref: our smooth reconstruction algorithm described in §4.2 ;
- FFDinit: the initial solution of our smooth reconstruction algorithm, as described in §4.2 ;
- Salz: the convex formulation proposed in [1]. This method is similar to SOCPimg except for the noise that is not handled the same way. In [1], the author minimizes a cost function that includes a 'reprojection error' in order to cope with the noise. In SOCPimg, the noise is handled with hard constraints.
- PerrioInit: the 'upper depth bound' approach of [3, 23] which is a point-wise algorithm that iteratively enforces the inextensibility constraints;
- PerrioRef: the 'refined approach' of [3, 23] which minimizes a cost function resulting in a refined estimation of the 3D points obtained with PerrioInit.

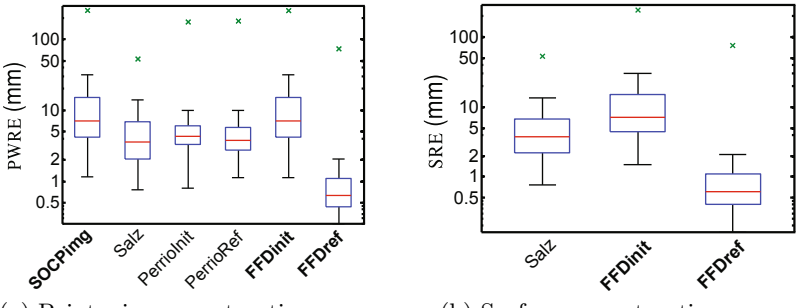
Reconstruction Errors. The discrepancy between the reconstructed and the ground truth surfaces are quantified with two measures, depending on the surface model used by the algorithms. The *point-wise reconstruction error* (PWRE), denoted e_p, can be used for all the algorithms. It is defined by:

$$e_p = \frac{1}{n_c} \sum_{i=1}^{n_c} \|\mathbf{Q}_i - \hat{\mathbf{Q}}_i\|. \tag{16}$$

For algorithms that uses more complex surface models, such as triangular meshes or FFD, we measure the *surface reconstruction error* (SRE), denoted e_s. It is the difference between the reconstructed surface $\mathcal{W}_{\boldsymbol{\ell}}$ and the ground truth surface $\hat{\mathcal{W}}$:

$$e_s = \iint \left\| \mathcal{W}_{\boldsymbol{\ell}}(\mathbf{q}) - \hat{\mathcal{W}}(\mathbf{q}) \right\| \mathrm{d}\mathbf{q}. \tag{17}$$

In this experiment, we use 1,000 randomly generated paper sheets with 150 points correspondences. Figure 3(a) shows the PWRE for all the algorithms and figure 3(b) shows the SRE for the algorithms that use a complex surface model. The main result of this experiment is that our approach FFDref gives the smallest reconstruction errors (PWRE and SRE). Globally, the methods that use complex surface models get better results than the point-wise approaches.



(a) Point-wise reconstruction error (b) Surface reconstruction error

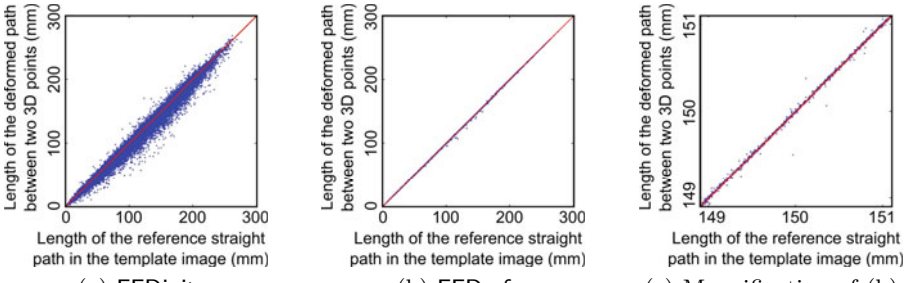
Fig. 3. Comparison of the reconstruction errors for different algorithms. The central red line is the median. The limits of the blue box are the 25th and the 75th percentiles. The black 'whiskers' cover approximately 99.3% of the experiment outcomes. The green crosses are the maximal errors over the 1000 trials.

Length of Geodesics. When a reconstructed 3D surface is reconstructed in a truly inextensible way, the transformation of the straight line linking two points in the template image must be the geodesic linking the corresponding two 3D points on the surface. In particular, the length of these two paths must be identical. Testing this hypothesis for our algorithms FFDinit and FFDref is the goal of this experiment. To do so, we use the same data as in the previous experiment. For each surface, we choose randomly 10,000 pairs of points in the template image. For each pair of points $(\mathbf{g}_i, \mathbf{g}_j)$, the length l_{ij}^{3D} of the deformed path linking the 3D points $\mathcal{W}_{\boldsymbol{\ell}}(\mathbf{g}_i)$ and $\mathcal{W}_{\boldsymbol{\ell}}(\mathbf{g}_j)$ on the surface is approximated by the length of the polygonal line linking these two points with the following formula:

$$l_{ij}^{3D} = \sum_{k=1}^{n_g} \left\| \mathcal{W}_{\boldsymbol{\ell}}\left(\mathbf{g}_i + \tfrac{k}{n_g}\|\mathbf{g}_j - \mathbf{g}_i\|\right) - \mathcal{W}_{\boldsymbol{\ell}}\left(\mathbf{g}_i + \tfrac{k-1}{n_g}\|\mathbf{g}_j - \mathbf{g}_i\|\right) \right\|, \tag{18}$$

where n_g is the number of intermediate points used for the approximation (we use $n_g = 200$ since we experimentally observed that the approximation stabilizes

for values of n_g greater than 180). The lengths of the deformed paths are plotted against their reference length in the template image in figure 4(a) for FFDinit and in figures 4(b,c) for FFDref. Figures 4(b) and 4(c) show that, with the surfaces reconstructed with FFDref, the length of the deformed paths are almost equal to the length they should have if they were actual geodesics. In other words, our approach FFDref reconstructs 3D surfaces which are truly inextensible. On the other hand, figure 4(a) shows that the initial solution FFDinit (which is just an FFD fitted to a sparse set of reconstructed 3D points) seems to be much less inextensible.



(a) FFDinit (b) FFDref (c) Magnification of (b)

Fig. 4. Plot of the length of deformed paths against the length they should have if the reconstructed surface was truly inextensible. The red diagonal line is the place where all the blue points should be for inextensible surfaces.

Let l_{ij}^{2D} be the Euclidean distance between the points $\mathbf{g}_i$ and $\mathbf{g}_j$. Table 2 gives some statistics on the relative error between the computed length l_{ij}^{3D} and the reference length l_{ij}^{2D}, *i.e.* the quantity $(l_{ij}^{2D} - l_{ij}^{2D})/l_{ij}^{3D}$. These numbers confirm the results seen in figure 4.

Table 2. Statistics on the relative errors between the length of transformed paths and the length they should have

	Mean	**Std deviation**	**Median**	**Minimum**	**Maximum**
FFDinit	0.0119	0.0417	0.0036	−1.9689	0.8931
FFDref	2.0084×10^{-5}	7.1965×10^{-4}	5.8083×10^{-6}	−0.0505	0.3396

Gaussian curvature. The Gaussian curvature is the product of the two principal curvature (which are the reciprocal of the radius of the osculating circle). For an inextensible surface, the Gaussian curvature is null. In this experiment, we check if this property is satisfied by the smooth surfaces reconstructed with FFDinit and FFDref. We used the same 1,000 reconstructed surfaces as in the previous experiment. The Gaussian curvature, denoted κ, is computed for 10,000 randomly chosen points on the surface with the formula $\kappa = \frac{\det(\mathbf{II})}{\det(\mathbf{I})}$, where $\mathbf{I}$

and **II** are the first and the second fundamental forms of the parametric surface [24]. The results of this experiment are reported in table 3. It shows that, in average, the Gaussian curvature of the surfaces reconstructed using FFDref are consistently close to 0. It also shows that FFDref gives Gaussian curvatures which are 100 times smaller than the ones obtained with FFDinit. These results demonstrate that the surfaces reconstructed with our approach FFDref are indeed inextensible. Note that this kind of experiment cannot be achieved if a smooth surface is not available.

Table 3. Statistics on the (absolute value of the) Gaussian curvatures for 1,000 reconstructed surfaces and 10,000 points per surface

	Mean	**Std deviation**	**Median**	**Minimum**	**Maximum**
FFDinit	4.9458×10^{-4}	0.0875	9.7302×10^{-5}	7.5122×10^{-14}	258.2379
FFDref	5.0046×10^{-6}	7.1320×10^{-4}	1.7333×10^{-6}	2.2325×10^{-14}	1.5199

5.2 Experiments on Real Data

The algorithms used in the synthetic experiments of §5.1 are applied to real data in figures 5 and 6. These figures show that our approaches give good results on real data. In particular, figure 5 shows that our method FFDref outperforms the other approaches in the presence of a self-occlusion. This comes from the fact that FFDref requires the surface to be inextensible everywhere, even if there are no point correspondences (which is the case on the self-occluded part of the paper sheet). An accurate stereo reconstruction of the surface in figure 6 were available. We compare in table 4 the average 3D errors between the surfaces reconstructed with a monocular approach to the stereo reconstruction. Again, our method FFDref is the one giving the best results.

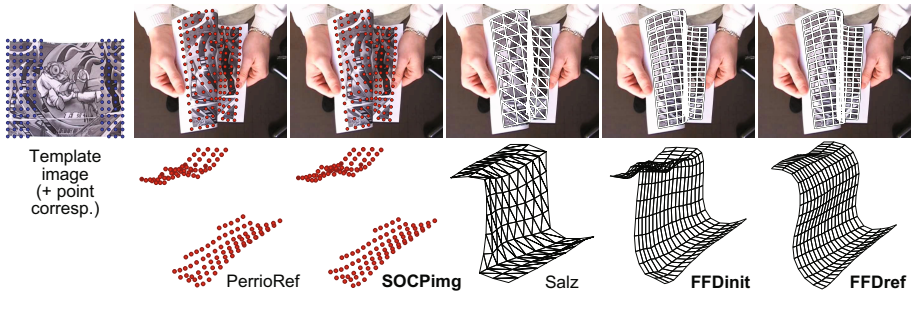


Fig. 5. Illustration of monocular reconstruction algorithms in the presence of a self-occlusion (the point correspondences were automatically extracted using [25]). Note how our algorithm FFDref is able to recover a reasonable shape for the occluded part.

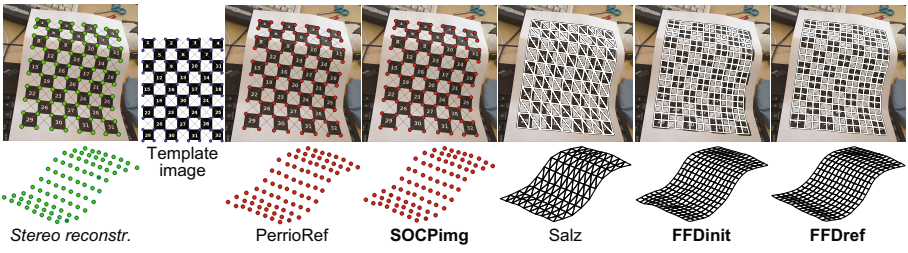


Fig. 6. Illustration of the results obtained with several monocular reconstruction algorithms. First row: input image along with a reprojection of the reconstructed 3D surface. Second row: reconstructed surface from a different point of view. Note that the stereo reconstruction (first column) is not a monocular algorithm: it is just used to assert the quality of the other reconstructed surfaces (see table 4).

Table 4. Average 3D error (in millimeters) with respect to the stereo reconstruction of the surface for the surfaces of figure 6

PerrioRef	SOCPimg	Salz	FFDinit	FFDref
2.388	2.261	4.743	2.259	**1.991**

6 Conclusion

In this paper, we presented new approaches for monocular reconstruction of inextensible surfaces imaged by a perspective camera. In particular, we proposed a SOCP formulation of the problem that accounts for noise in both the template and the input images. We also designed an algorithm that directly reconstruct a smooth surface based on free-form deformations. This algorithm outperforms previous approaches in terms of precision of the reconstructed surface. Besides, we experimentally showed that the surfaces reconstructed with this algorithm are truly inextensible. The only drawback of this approach is that it is formulated as a non-linear least-squares minimization problem with a non-convex cost function. However, we proposed a method to build an initial solution which is close to the optimum. It allows us to get rid of the difficulties linked to the non-convexity of the cost function.

Acknowledgement. This work has been partly founded by the Regional Council of Auvergne. NICTA is funded by the Australian Government, in part through the Australian Research Council.

References

1. Salzmann, M., Fua, P.: Reconstructing sharply folding surfaces: A convex formulation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1054–1061 (2009)

2. Shen, S., Shi, W., Liu, Y.: Monocular 3-D tracking of inextensible deformable surfaces under L_2-norm. IEEE Transactions on Image Processing 19, 512–521 (2010)
3. Perriollat, M., Hartley, R., Bartoli, A.: Monocular template-based reconstruction of inextensible surfaces. International Journal of Computer Vision (2010)
4. Bregler, C., Hertzmann, A., Biermann, H.: Recovering non-rigid 3D shape from image streams. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2690–2696 (2000)
5. Bartoli, A., Gay-Bellile, V., Castellani, U., Peyras, J., Olsen, S., Sayd, P.: Coarse-to-fine low-rank structure-from-motion. In: IEEE Conference on Computer Vision and Pattern Recognition (2008)
6. Brand, M.: A direct method for 3D factorization of nonrigid motion observed in 2D. In: IEEE Conference on Computer Vision and Pattern Recognition (2005)
7. Del Bue, A.: A factorization approach to structure from motion with shape priors. In: IEEE Conference on Computer Vision and Pattern Recognition (2008)
8. Olsen, S., Bartoli, A.: Implicit non-rigid structure-from-motion with priors. Journal of Mathematical Imaging and Vision 31, 233–244 (2008)
9. Torresani, L., Hertzmann, A., Bregler, C.: Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. IEEE Transactions on Pattern Analysis and Machine Intelligence 30, 878–892 (2008)
10. Xiao, J., Chai, J., Kanade, T.: A closed-form solution to non-rigid shape and motion recovery. International Journal of Computer Vision 67, 233–246 (2006)
11. Gay-Bellile, V., Perriollat, M., Bartoli, A., Sayd, P.: Image registration by combining thin-plate splines with a 3D morphable model. In: International Conference on Image Processing (2006)
12. Salzmann, M., Hartley, R., Fua, P.: Convex optimization for deformable surface 3-D tracking. In: IEEE International Conference on Computer Vision (2007)
13. Salzmann, M., Urtasun, R., Fua, P.: Local deformation models for monocular 3D shape recovery. In: IEEE Conference on Computer Vision and Pattern Recognition (2008)
14. Gumerov, N., Zandifar, A., Duraiswami, R., Davis, L.S.: Structure of applicable surfaces from single views. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3023, pp. 482–496. Springer, Heidelberg (2004)
15. Prasad, M., Zisserman, A., Fitzgibbon, A.W.: Single view reconstruction of curved surfaces. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 1345–1354 (2006)
16. 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)
17. Shen, S., Shi, W., Liu, Y.: Monocular template-based tracking of inextensible deformable surfaces under l_2-norm. In: Zha, H., Taniguchi, R.-i., Maybank, S. (eds.) ACCV 2009. LNCS, vol. 5995, pp. 214–223. Springer, Heidelberg (2010)
18. Rueckert, D., Sonoda, L., Hayes, C., Hill, D., Leach, M., Hawkes, D.: Nonrigid registration using free-form deformations: Application to breast MR images. IEEE Transactions on Medical Imaging 18, 712–721 (1999)
19. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
20. Zhu, J., Hoi, S., Lyu, M.: Nonrigid shape recovery by gaussian process regression. In: IEEE Conference on Computer Vision and Pattern Recognition (2009)
21. Dierckx, P.: Curve and Surface Fitting with Splines. Oxford University Press, Oxford (1993)

22. Perriollat, M., Bartoli, A.: A quasi-minimal model for paper-like surfaces. In: Proceedings of the ISPRS International Workshop Towards Benmarking Automated Calibration, Orientation, and Surface Reconstruction from Images (2007)
23. Perriollat, M., Hartley, R., Bartoli, A.: Monocular template-based reconstruction of inextensible surfaces. In: British Machine Vision Conference (2008)
24. Gray, A.: The Gaussian and Mean Curvatures. In: Modern Differential Geometry of Curves and Surfaces with Mathematica, pp. 373–380. CRC Press, Boca Raton (1997)
25. Gay-Bellile, V., Bartoli, A., Sayd, P.: Direct estimation of non-rigid registrations with image-based self-occlusion reasoning. IEEE Transactions on Pattern Analysis and Machine Intelligence 32, 87–104 (2008)

Multi-class Leveraged k-NN for Image Classification

Paolo Piro[1], Richard Nock[2], Frank Nielsen[3], and Michel Barlaud[1]

[1] University of Nice-Sophia Antipolis / CNRS, France
[2] CEREGMIA, University of Antilles-Guyane, France
[3] Sony CSL / LIX, Ecole Polytechnique, France

Abstract. The k-nearest neighbors (k-NN) classification rule is still an essential tool for computer vision applications, such as scene recognition. However, k-NN still features some major drawbacks, which mainly reside in the *uniform voting* among the nearest prototypes in the feature space.

In this paper, we propose a new method that is able to learn the "relevance" of *prototypes*, thus classifying test data using a *weighted* k-NN rule. In particular, our algorithm, called Multi-class Leveraged k-nearest neighbor (MLNN), learns the prototype weights in a *boosting* framework, by minimizing a *surrogate* exponential risk over training data. We propose two main contributions for improving computational speed and accuracy. On the one hand, we implement learning in an inherently *multiclass* way, thus providing significant computation time reduction over one-versus-all approaches. Furthermore, the leveraging weights enable effective data selection, thus reducing the cost of k-NN search at classification time. On the other hand, we propose a *kernel* generalization of our approach to take into account real-valued similarities between data in the feature space, thus enabling more accurate estimation of the local class density.

We tested MLNN on three datasets of natural images. Results show that MLNN significantly outperforms classic k-NN and weighted k-NN voting. Furthermore, using an adaptive Gaussian kernel provides significant performance improvement. Finally, the best results are obtained when using MLNN with an appropriate learned metric distance.

1 Introduction

In this paper, we address the task of image categorization. This task aims at automatically classifying images into a predefined set of scene *categories*, like the natural scenes represented in Fig. 1. (See Sec. 3.1 for a detailed description of the databases we used in our experiments). Despite lots of works, much remains to be done to challenge human level performances, not only because there is a huge number of natural categories that should be considered in general. In fact, images carry only parts of the information that is used by humans to categorize, and parts of the information available from images may be highly misleading: for example, natural image categories may exhibit high *intra-class* variability

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 67–81, 2011.

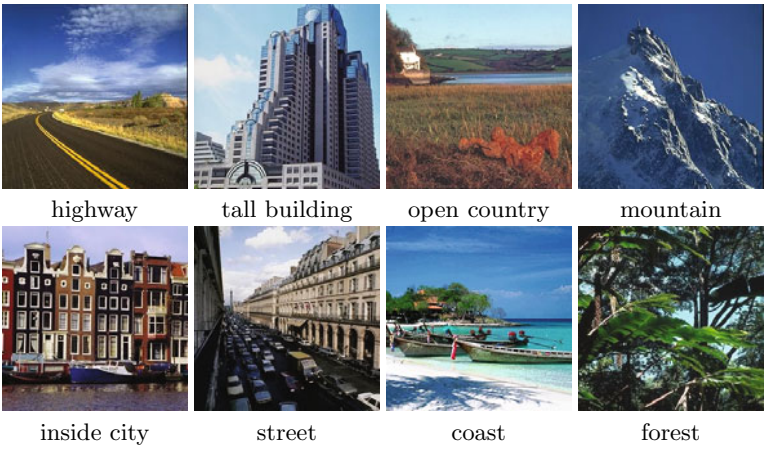


Fig. 1. The first dataset we used in our experiments consists of 8 categories of natural scenes [1]

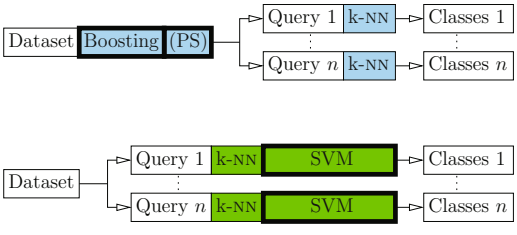


Fig. 2. Optimizing k-NN via MLNN (up, blue) and SVM-KNN [2] (down, green). MLNN uses a boosting algorithm before being presented any query, while SVM-KNN learns support vectors after each query is presented. Bold rectangles indicate induction steps (PS = prototype selection; see text for details).

(*i.e.*, visually different images may belong to the same category) and low *inter-class* variability (*i.e.*, distinct categories may contain images that are visually similar).

For the purpose of automatic classification of images, the k-NN classification has shown to be very effective [3]. Its use is supported by a wide spectrum of arguments, ranging from the field of philosophy to that of mathematics [2]. The simplicity of the method — use the labeled neighbor(s) of a query to predict its class — makes it a good candidate for further improvements, desirable in part because of statistical and computational drawbacks [2]. So far, the literature has favoured two main ways to cope with these issues: improve classification accuracy by means of local classifiers [2,4,5,6], or filter out ill-defined examples [7], with intermediate approaches [8]. Many of these algorithms can be viewed as primers to improve the (continuous) estimation of class membership probabilities [9], but none has

completely succeeded in this task. This problem has been reformulated by Marin et al. [10] as a strong advocacy for the formal transposition of *boosting* to k-NN classification. This issue is challenging as k-NN rules are indeed not induced, whereas formal boosting algorithms combine *two* induction steps, inducing so-called *strong* classifiers by combining *weak* classifiers (also induced). Previously, Athitsos and Sclaroff [11] had already proposed an approach to bring the boosting principle into the k-NN classification framework. However, their method consisted in "boosting" the distance measure, *i.e.*, learning a combination of metric distances that could improve the generalization of classifier. Furthermore, their classification framework was not intrinsically multiclass, as they formulated the problem as binary learning on random triplets of training data.

In this paper, we tackle the issue of integrating k-NN in a boosting framework from a different perspective. In particular, our algorithm, called MLNN, induces a multiclass leveraged k-nearest neighbor rule that generalizes the uniform k-NN rule, using the examples directly as weak hypotheses (that we also call *prototypes*). Our MLNN method does not need to learn a distance function, as it directly operates on the top of k-nearest neighbors search. Furthermore, it does not require an explicit computation of the feature space, thus preserving one of the main advantages of prototype-based methods. Compared to the well-known SVM-k-NN local learning approach [2], MLNN also speeds up query processing: instead of learning a local classifier for each query, MLNN performs learning upwards, once and for all, and does not need to be run again or updated depending on queries (Fig. 2). Finally, the most significant advantage of MLNN lies in its ability to find out the most relevant prototypes for categorization, allowing to filter out the remaining less reliable examples. Experimentally, significant data reductions are observed with a simultaneous increase in categorization performances.

In Sec. 2 we present our MLNN approach, along with the statement of its theoretical properties. In order not to laden the paper's body, the proofsketches of the results have been postponed to an appendix. Then, Sec. 3 displays the behavior of MLNN on three standard databases of real-world image categorization. Finally, we conclude with some observations (Sec. 4).

2 Method

2.1 Problem Statement and Notations

In this paper, we address the task of multiclass image categorization. It consists in assigning an image to one of several predefined categories (or classes, or labels). Instead of splitting the multiclass problem in as many *one-versus-all* (binary) classification problems — a frequent approach in boosting [12] — we directly tackle the *multiclass* problem, following Zou et al [13]. For a given query image, we compute its *classification score* for all categories. While we basically use this vector for single-label prediction using the category with the maximum score, our algorithm can be straightforwardly extended to multilabel prediction and ranking [12]. We suppose given a set $\mathcal{S}$ of m annotated images. Each image is

a training *example* $(\boldsymbol{x}, \boldsymbol{y})$, where $\boldsymbol{x}$ is the image feature vector and $\boldsymbol{y}$ the *class vector* that specifies the category membership of the image. In particular, the sign of component y_c gives the positive/negative membership of the example to class c ($c = 1, 2, ..., C$). Inspired by the multiclass boosting analysis of Zou et al [13], we constrain the class vector to be *symmetric*, *i.e.*: $\sum_{c=1}^{C} y_c = 0$, by setting: $y_{\tilde{c}} = 1$, $y_{c \neq \tilde{c}} = -\frac{1}{C-1}$, where $\tilde{c}$ is the true image category. Furthermore, we denote by $K(\boldsymbol{x_i}, \boldsymbol{x_j})$ a symmetric similarity kernel on the pair of examples $\boldsymbol{x_i}$, $\boldsymbol{x_j}$.

2.2 (Leveraged) *k*-Nearest Neighbors

The vanilla k-NN rule is based on majority vote among the k-nearest neighbors in set $\mathcal{S}$, to predict the class of query $\boldsymbol{x}_q$. It can be defined as the following multiclass classifier $\boldsymbol{h} = \{h_c,\ c = 1, 2, ..., C\}$:

$$h_c(\boldsymbol{x}_q) = \frac{1}{k} \sum_{i \sim_k q} [y_{ic} > 0] \quad , \tag{1}$$

where $h_c \in [0, 1]$ is the classification score for class c, $i \sim_k q$ denotes an example $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ belonging to the k-nearest neighbors of $\boldsymbol{x}_q$ and square brackets denote the indicator function.

In this paper, we propose to generalize (1) to the following *leveraged* k-NN rule $\boldsymbol{h}^\ell = \{h_c^\ell\}$:

$$h_c^\ell(\boldsymbol{x}_q) = \sum_{j=1}^{T} \alpha_j K(\boldsymbol{x}_q, \boldsymbol{x}_j) y_{jc} \in \mathbb{R} \quad , \tag{2}$$

where prediction h_c^ℓ takes values in all $\mathbb{R}$. In (2), we have introduced the three following elements to generalize (1):

- *leveraging coefficients* α_j, that provide a *weighted* voting rule instead of uniform voting;
- kernel K, which takes into account "soft" (real-valued) similarities between query $\boldsymbol{x}_q$ and prototypes $\boldsymbol{x}_j$, instead of "hard" selection of the most similar (k-NN) prototypes;
- size T of the set of *prototypes* that are allowed to vote.

This last point is particularly interesting for computational purposes, as our classification rule actually involves only a (possibly sparse) subset of the training data as prototypes to be used at query time. Indeed, a *prototype selection* step is to be performed while training our classifier, in order to determine the most relevant subset of training data, *i.e.*, the so-called *prototypes*, forming a set $\mathcal{P} \subseteq \mathcal{S}$ (Figure 2). The prototypes are selected during the training phase, which consists in fitting their coefficients α_j, while removing the least relevant annotated data from $\mathcal{S}$.

2.3 Multiclass Surrogate Risk Minimization

In order to fit our leveraged classification rule (2) onto training set $\mathcal{S}$, we focus on the minimization of a multiclass surrogate[1] (exponential) risk:

$$\varepsilon^{\exp}\left(\boldsymbol{h}^{\ell}, \mathcal{S}\right) \doteq \frac{1}{m}\sum_{i=1}^{m}\exp\left\{-\rho(\boldsymbol{h}^{\ell}, i)\right\} \ , \tag{3}$$

where:

$$\rho(\boldsymbol{h}^{\ell}, i) = \frac{1}{C}\sum_{c=1}^{C} y_{ic} h_c^{\ell}(\boldsymbol{x}_i) \tag{4}$$

is the multiclass *edge* of classifier $\boldsymbol{h}^{\ell}$ on training example $\boldsymbol{x}_i$. In particular, this edge averages over the C classes the "goodness of fit" of classifier $\boldsymbol{h}^{\ell}$ on example $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, thus being positive iff the prediction agrees with the example's annotation. Therefore, counting the number of negative edges enables to quantify the so-called *empirical risk*, *i.e.*, the actual misclassification rate on the training data, as follows:

$$\varepsilon^{0/1}\left(\boldsymbol{h}^{\ell}, \mathcal{S}\right) \doteq \frac{1}{m}\sum_{i=1}^{m}\left[\rho(\boldsymbol{h}^{\ell}, i) < 0\right] \ . \tag{5}$$

Rather than directly tackling the problem of minimizing $\varepsilon^{0/1}$ — which is not differentiable and often computationally hard to minimize [14] — we concentrate on the optimization of surrogate (3), which is an *upper bound* of the empirical risk.

In order to solve this optimization, we propose a boosting-like procedure, *i.e.*, an iterative strategy where the classification rule is updated by adding a new prototype $(\boldsymbol{x}_j, \boldsymbol{y}_j)$ (weak classifier) at each step t ($t = 1, 2, \ldots, T$), thus updating the strong classifier (2) as follows:

$$h_c^{(t)}(\boldsymbol{x}_i) = h_c^{(t-1)}(\boldsymbol{x}_i) + \delta_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) y_{jc} \ . \tag{6}$$

(j is the index of the prototype chosen at iteration t.) Using (6) into (4), and then plugging it into (3), turns the problem of minimizing (3) to that finding δ_j with the following objective:

$$\arg\min_{\delta_j} \sum_{i=1}^{m} w_i \cdot \exp\left\{-\delta_j \mathrm{r}_{ij}\right\} \ . \tag{7}$$

In (7), we have defined w_i as the weighting factor, depending on the past weak classifiers:

$$w_i = \exp\left\{-\frac{1}{C}\sum_{c=1}^{C} y_{ic} h_c^{(t-1)}(\boldsymbol{x}_i)\right\} \ , \tag{8}$$

[1] We call *surrogate* a function that upperbounds the risk functional we should minimize, and thus can be used as a primer for its minimization.

and r_{ij} as a pairwise term only depending on training data:

$$\mathrm{r}_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)\frac{1}{C}\sum_{c=1}^{C} y_{ic}y_{jc} \ . \tag{9}$$

Finally, taking the derivative of (7), the global minimization of surrogate risk (3) amounts to fitting δ_j so as to solve the following equation:

$$\sum_{i=1}^{m} w_i \mathrm{r}_{ij} \exp\left\{-\delta_j \mathrm{r}_{ij}\right\} = 0 \ . \tag{10}$$

Algorithm 1. MULTI-CLASS LEVERAGED k-NN MLNN ($\mathcal{S}$)

Input: $\mathcal{S} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i) \ , \quad i = 1, 2, ..., m \ , \quad \boldsymbol{y_i} \in \{-\frac{1}{C-1}, 1\}^C\}$

Let $\mathrm{r}_{ij} \doteq \frac{1}{C}\sum_{c=1}^{C} K(\boldsymbol{x}_i, \boldsymbol{x}_j) y_{ic} y_{jc}$ (11)

Let $\alpha_j \leftarrow 0, \quad \forall j = 1, 2, ..., m$

Let $w_i \leftarrow 1/m, \quad \forall i = 1, 2, ..., m$

for $t = 1, 2, ..., T$ **do**

  [I.0] Weak index chooser oracle:

  Let $j \leftarrow \textsc{Wic}(\{1, 2, ..., m\}, t)$;

  [I.1] Compute δ_j solution of:

$$\sum_{i=1}^{m} w_i \mathrm{r}_{ij} \exp\left\{-\delta_j \mathrm{r}_{ij}\right\} = 0 \ ; \tag{12}$$

  [I.2] Let

$$w_i \leftarrow w_i \exp(-\delta_j \mathrm{r}_{ij}), \quad \forall i \ : \ j \sim_k i \ ; \tag{13}$$

  [I.3] Let $\alpha_j \leftarrow \alpha_j + \delta_j$

Output: $h_c^{\ell}(\boldsymbol{x}_q) = \sum_{j \sim_k q} \alpha_{jc} y_{jc}, \quad \forall c = 1, \ldots, C$

2.4 MLNN: Multi-class Leveraged k-NN Rule

Pseudocode of MLNN is shown in Alg. 1. The main ingredient to compute leveraging coefficients relies on the so-called *edge matrix* R with general entry $\mathrm{r}_{i,j}$ (Eq. 9). This term depends on the pairwise similarity between two training examples, as it is given by the kernel, as well as on the ground-truth annotations. Indeed, it combines a "labeling" term, which determines the sign, *i.e.*, being positive iff labels of i and j agree, with a "geometric" term, which influences the magnitude, *i.e.*, being larger when the two examples are closer to each other in the feature space.

We distinguish the following two cases, depending on which kernel is selected:

k-NN kernel. In the most basic setting, *i.e.*, when using k-NN kernel, term $K(\boldsymbol{x}_i, \boldsymbol{y}_i)$ behaves like an indicator function that only selects the k-NN of i. Therefore, in this case (9) simplifies to:

$$\mathrm{r}_{ij} \doteq \begin{cases} \frac{1}{C}\sum_{c=1}^{C} y_{ic}y_{jc} & \text{if } j \sim_k i \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

and (10) has the following closed-form solution:

$$\delta_j \leftarrow \frac{(C-1)^2}{C}\log\left(\frac{(C-1)w_j^+}{w_j^-}\right), \tag{15}$$

with:

$$w_j^+ = \sum_{i:\ \mathrm{r}_{ij}>0} w_i, \quad w_j^- = \sum_{i:\ \mathrm{r}_{ij}<0} w_i\ . \tag{16}$$

general kernel. When using any kernel, entries of the edge matrix (Eq. 9) are real-valued and, in general, not sparse. Moreover, the equation (10) is transcendental, thus not admitting a closed-form solution. Hence, we compute the solution numerically, implementing a Newton's iterative method. This method gives the following approximation at step $k+1$, given the previous one at step k:

$$\delta^{(k+1)} = \delta^k + \frac{\sum_{i=1}^{m} w_i \mathrm{r}_{ij} \exp\{-\delta^{(k)}\mathrm{r}_{ij}\}}{\sum_{i=1}^{m} w_i \mathrm{r}_{ij}^2 \exp\{-\delta^{(k)}\mathrm{r}_{ij}\}}\ . \tag{17}$$

A critical setting for obtaining quick convergence of the solution is the initialization. Here, we propose to initialize the algorithm with the root of a linearized version of Eq. (10):

$$\delta^{(0)} = \frac{\sum_{i=1}^{m} w_i \mathrm{r}_{ij}}{\sum_{i=1}^{m} w_i \mathrm{r}_{ij}^2}\ . \tag{18}$$

A suitable choice for the kernel is the Radial Basis Function (RBF), which provides "smooth" pairwise similarities between feature points:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left\{-\frac{||\boldsymbol{x}_i - \boldsymbol{x}_j||^2}{2\sigma^2}\right\}, \tag{19}$$

where parameter σ may be either constant or adapted to the local sample density (*e.g.*, one may set $\sigma = \rho_k(\boldsymbol{x}_i)$, where $\rho_k(\boldsymbol{x}_i)$ is the k-NN distance to $\boldsymbol{x}_i$, thus "enlarging" the window size where training data are sparser.) In particular, in the following experiments we use a Gaussian kernel that is truncated to the first k nearest neighbors, thus providing a straightforward generalization of the k-NN kernel. In this case the edge matrix writes as follows:

$$\mathrm{r}_{ij} \doteq \begin{cases} \frac{1}{C}\sum_{c=1}^{C} \exp\left\{-\frac{||\boldsymbol{x}_i-\boldsymbol{x}_j||^2}{2\sigma^2}\right\} y_{ic}y_{jc} & \text{if } j \sim_k i \\ 0 & \text{otherwise} \end{cases}\ . \tag{20}$$

Another ingredient of MLNN is more common to boosting algorithms: MLNN operates on a set of weights w_i $(i = 1, 2, ..., m)$ defined over training data. These

weights are repeatedly updated, such that those of mislabelled examples are increased, and vice-versa.

At each iteration t of the algorithm, a *weak index chooser* oracle $\textsc{Wic}(\{1,2,...,m\},t)$ determines index $j \in \{1,2,...,m\}$ of the example to leverage (step I.0). Various choices are possible for this oracle. The simplest is perhaps to compute Eq. (16, 12) for all the training examples. δ_j in Eq. (12) can indeed be used to obtain a local measure of the class density [14], which is as better as δ_j gets large. This simple oracle thus picks j maximizing δ_j:

$$j \leftarrow \textsc{Wic}(\{1,2,...,m\},t) \ : \ \delta_j = \max_{j \in \{1,2,...,m\}} \delta_j^t \ . \tag{21}$$

This oracle allows an example to be chosen more than once, thus letting its leveraging coefficient α_j be updated several times (step I.3). It is known that, in order to be statistically consistent, some boosting algorithms require to be run for $T \ll m$ rounds [15]. Cast in the setting of MLNN, this constraint precisely supports prototype selection, as T is an upperbound for the number of examples with non-zero leveraging coefficients.

MLNN shares the property with boosting algorithms of being resources-friendly: since computing the leveraging coefficients scales linearly with the number of neighbors, the time complexity bottleneck of MLNN does not rely on boosting, but rather on the complexity of k-NN search. Furthermore, notice that, when whichever w_j^+ or w_j^- is zero, δ_j in (12) is not finite. There is a simple way to eliminate this drawback, inspired by [12]: we add $1/m$ to both the numerator and the denominator of the fraction in the log term of (12). This smoothes out δ_j, guaranteeing its finiteness without impairing convergence of MLNN.

In the following section, we provide formal details about the boosting analysis of MLNN.

2.5 Properties of MLNN

Two fundamental theorems hold for MLNN.

Theorem 1. MLNN *converges with T to $\boldsymbol{h}^\ell$ realizing the* ***global*** *minimum of the exponential risk (3).*

MLNN is a specialization of a very general learning algorithm which keeps the same convergence guarantee when replacing the surrogate risk (3) by elements of a broad class of surrogates risk [14,15].

The second theorem provides a convergence rate for MLNN, which is based on a fundamental assumption on weak classifiers.

Theorem 2. *Let $p_j \doteq w_j^+/(w_j^+ + w_j^-)$. If the following weak index assumption (**WIA**) holds for $\tau \leq T$ steps in* MLNN*:*

(WIA). *There exist some $\gamma > 0$ and $\eta > 0$ such that the following two inequalities hold for index j returned by* $\textsc{Wic}(\{1,2,...,m\},t)$*:*

$$|p_j - \frac{1}{C}| \geq \gamma \ , \tag{22}$$

$$(w_j^+ + w_j^-)/||\boldsymbol{w}||_1 \geq \eta \ . \tag{23}$$

Then: $\varepsilon^{0/1}(\boldsymbol{h}^\ell, \mathcal{S}) \leq \exp(-\frac{C}{C-1}\eta\gamma^2\tau)$.

(Proofsketch in appendix) Ineq. (22) is the usual weak learning assumption, used to analyze classical boosting algorithms [16,12], when considering examples as weak classifiers. A *weak coverage assumption* (23) is needed as well, because insufficient *coverage* of the reciprocal neighbors could easily wipe out the surrogate risk reduction due to a large γ in (22). In the framework of k-NN classification, choosing k not too small is enough for the **WIA** to be met for a large number of boosting rounds τ, thus determining a potential harsh decrease of $\varepsilon^{\exp}(\boldsymbol{h}^\ell, \mathcal{S})$. This is important, as a big difference with classical boosting algorithms (*e.g.* AdaBoost [16]) is that oracle WIC(.,.) has only access to m different weak classifiers, *i.e.*, one per example. Finally, the bound in Theorem 2 shows that classification (22) may be more important than coverage (23) for nearest neighbors.

3 Experiments

In this section, we present experimental results of MLNN with different kernel settings and comparison with both k-NN and ITML [6], which is a state-of-the-art metric learning algorithm. In particular, our experiments aim at evaluating the effect of UNN sparse prototype selection on the classification accuracy. For this purpose, we measured the classification performances when varying the number of prototypes retained at test time. In MLNN, prototype selection is carried out by setting $T < m$, which corresponds to retaining at most T relevant prototypes. When running the other methods, we carried out random prototype selection and averaged the results over a number of iterations.

3.1 Scene Categorization

We validated our MLNN algorithm on three well-known image categorization databases.

8-cat: firstly proposed by [1], includes 2,688 color images grouped into eight categories: 360 coast, 328 forest, 374 mountain, 410 open country, 260 highway, 308 inside of cities, 356 tall buildings, and 292 street (Fig. 1).

13-cat: adds five more categories of gray-scale images to the 8-cat database [17]: 241 suburb residence, 174 bedroom, 151 kitchen, 289 living room, and 216 office.

15-cat: includes 13-cat database plus two more categories (gray-scale images) [18]: 315 store, and 311 industrial.

In the following section we report results obtained by splitting each database in two distinct subsets, one for training, the other for test. We always used about 2,000 randomly selected training images. Namely, 250 images per category were

selected from the 8-cat database, 150 from the other two datasets. The remaining images were used for testing. In our experiments, we mostly concentrated on evaluating the trade-off between classification accuracy and computational time, as provided by selecting a sparse prototype dataset from the training data. In particular, fixing the number of prototypes amounts to fixing the computational cost of classification, as this latter only depends on the cost of k-NN search on the prototype set. (So as for k-NN, a random sample of the training set was selected and results were averaged over a number of random sampling realizations.) All the results we present were obtained with $k = 11$ and pre-processing Gist features [1] with PCA down to dimension $d = 128$.

We compared different implementations of our MLNN algorithm. In particular, we tested:

- MLNN with the basic setting, *i.e.*, the uniform k-NN kernel of Eq. (14);
- WMLNN, *i.e.*, MLNN with fixed-size Gaussian kernel (19) (with $\sigma = 0.25$);
- AdaWMLNN, *i.e.*, MLNN with adaptive-size Gaussian kernel (19) (with $\sigma = \sqrt{2}\rho_k(\boldsymbol{x}_i)$, $\rho_k(\boldsymbol{x}_i)$ being the k-NN distance from example $\boldsymbol{x}_i$);
- MLNN "one-versus-all", *i.e.* Alg. 1 with $C = 2$ applied to each category independently (considering examples in the current category as "positives", the remaining ones as "negatives").

Furthermore, we compared our method with different k-NN–based classification methods, which either rely on metric learning or not. Namely, we tested:

- classic non-parametric k-NN voting;
- weighted k-NN (Wk-NN) voting with Gaussian weights, as proposed by Philbin et al. [19]; we used (19) with $\sigma = 1$ as a weighting factor;
- k-NN voting combined with ITML metric learning [6].

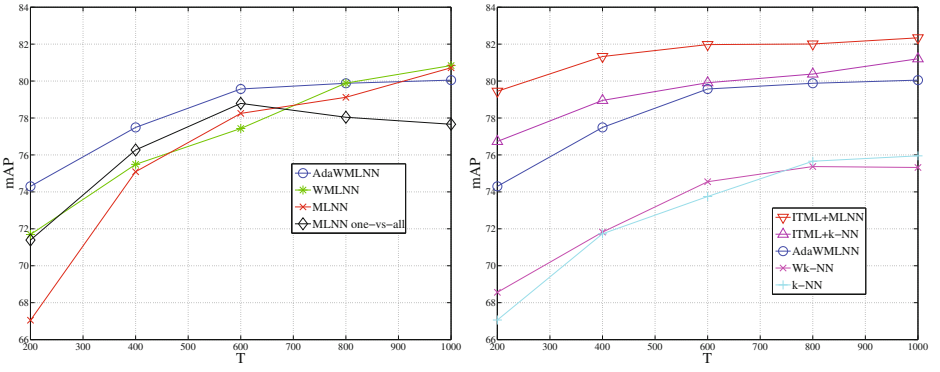
We tested all these methods for a fixed number of prototypes, *i.e.*, for a fixed computational cost of classification. In particular, a random sample of the training set was selected and results were averaged over a number of random sampling realizations.

Finally, we integrated the ITML method with MLNN in order to provide a unique method for addressing simultaneously both the choice of the metric distance and the rejection of "noisy" examples, which are the two fundamental issues of k-NN classification.

3.2 Categorization Results

The categorization test consists in assigning each test image to one of the predefined categories. We measured the overall performance rate as the mean Average Precision (mAP), which is the average of the classification rates for each category.

In Fig. 3(a) we compare the results of MLNN with the abovementioned settings. Interestingly, these results show the significant improvement provided by using a "smooth" kernel for learning the prototypes. Namely, the adaptive-size kernel provides the best performances. Furthermore, the gap over the basic



(a) (b)

Fig. 3. Experimental results of categorization on 8-cat database in terms of mAP as a function of the number of prototypes, for $k = 11$ and Gist descriptors of dimension $d = 128$ (after PCA). (a) Comparison between 3 different implementations of MLNN and one-versus-all MLNN. (b) Comparison between MLNN with adaptive Gaussian kernel (AdaWMLNN), k-NN, weighted k-NN and MLNN one-versus-all (UNN).

MLNN is more consistent when retaining less prototypes, as AdaMLNN enables a finer class density estimation even with very sparse examples (see, for instance, the performance gap of 7% between MLNN and AdaWMLNN for $T = 200$). Furthermore, notice that the multiclass version of our algorithm outperforms the one-versus-all implementation (gap between 1% and 3%). Hence, our multiclass MLNN not only is much less computationally expensive than one-versus-all MLNN, as it avoids to run the boosting procedure C times independently, but also provides better classification accuracy.

On the same 8-cat database we compared AdaWMLNN to k-NN voting with or without metric learning (Fig. 3(b)). First of all, we notice that our AdaWMLNN method significantly outperforms k-NN and Wk-NN, *i.e.,* non-learned voting rules (up to 6% improvement). Then, performances of our method are overall comparable to those of ITML, being slightly inferior to them, but the computational cost of MLNN is considerably lower than that of metric learning. Finally, our results show that, when combined with a metric learning strategy, MLNN is able to significantly outperform all the other classification methods, thus enabling a significant accuracy improvement over the state-of-the-art (up to 3% when retaining few prototypes, see for example performance at $T = 200$).

In Fig. 4 we focus on a more extensive comparison between regular MLNN and classic k-NN on the 13-cat and 15-cat datasets. Here, we report the mean Average Precision as a function of the number of prototypes per category. (Since this number varies from category to category, we report the average number of prototypes over all categories.) Notice that the gap between the two methods is most significant when retaining less than half prototypes, namely 6% improvement with 80 prototypes on 8-cat database, 7% with only 60 prototypes for both 13-cat and 15-cat. Besides considerably improving precision over k-NN, we also

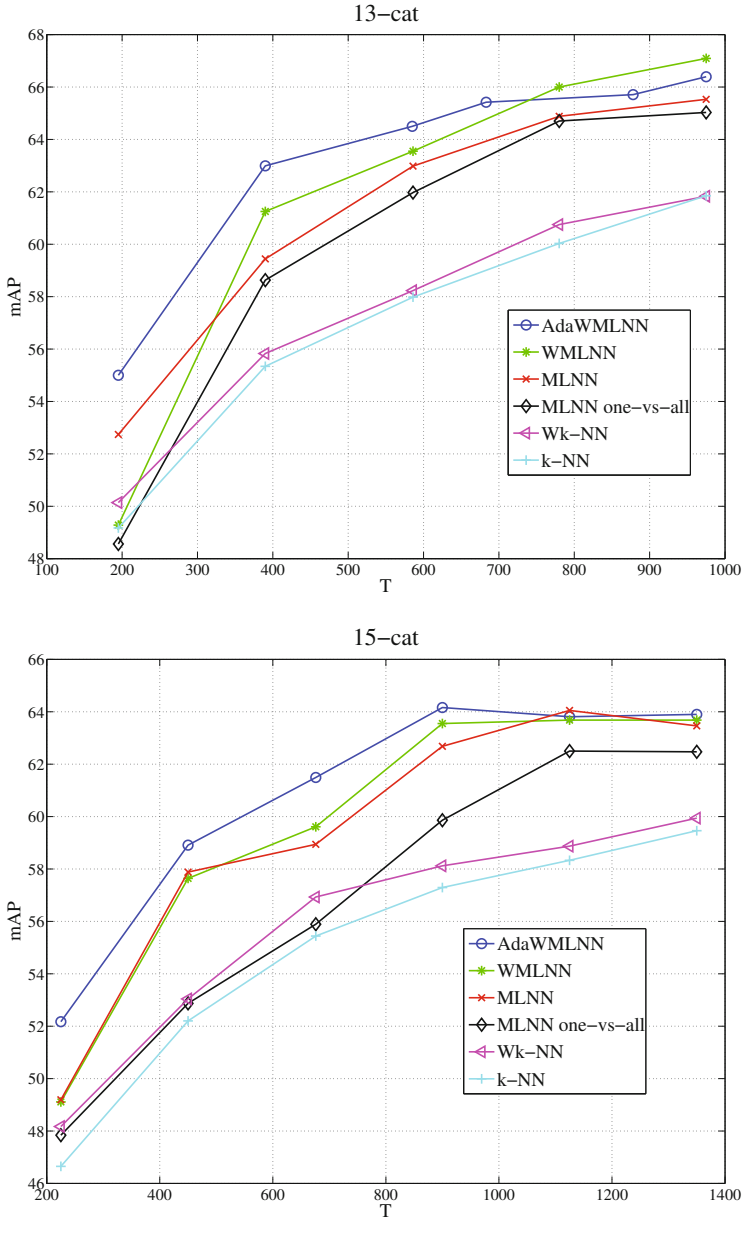


Fig. 4. Performance of MLNN with different settings (see the paper for details) compared to k-NN and weighted k-NN as a function of the number of prototypes per class for 13-cat (4(a)) and 15-cat (4(b)) datasets.

drastically reduce the computational complexity of classification, which deals with finding nearest neighbors on a sparse dataset (gain up to a factor 4 when discarding half prototypes).

4 Conclusion

In this paper, we have proposed a novel boosting algorithm, MLNN, which learns a leveraged k-NN rule following the minimization of a multiclass surrogate (exponential) risk. This rule generalizes k-NN to weighted voting. Under mild learning and coverage assumptions, MLNN convergence is proven to be exponentially fast. Experiments on benchmark image categorization databases display that MLNN is significantly more accurate than k-NN (up to 6%), achieving very significant improvement on image databases with only few thousand images. Since the number of weak hypotheses available for boosting is in the order of the number of images, improvements may rapidly become dramatic as databases get larger. MLNN also provides us with a very simple and efficient prototype selection method reducing the cost of searching for neighbors at classification time. MLNN precision is comparable with that of a state-of-the-art metric learning method [6]. Since MLNN is still fully compatible with any underlying distortion and data structure for k-NN search, it also take advantage from learning a metric distance, thus simultaneously solving both major issues of k-NN voting: selection of a suitable metric distance and rejection of "noisy" prototypes. Last but not least, MLNN is simple and modular enough, so that it can be easily extended to work on global image descriptors (like Bags of Features, Fisher Kernels ...)

References

1. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. Int. J. of Comp. Vision 42, 145–175 (2001)
2. Zhang, H., Berg, A.C., Maire, M., Malik, J.: Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In: CVPR 2006, pp. 2126–2136 (2006)
3. Boiman, O., Shechtman, E., Irani, M.: In defense of nearest-neighbor based image classification. In: CVPR 2008, pp. 1–8 (2008)
4. Hastie, T., Tibshirani, R.: Discriminant adaptive nearest neighbor classification. IEEE Trans. PAMI 18, 607–616 (1996)
5. Paredes, R.: Learning weighted metrics to minimize nearest-neighbor classification error. IEEE Trans. PAMI 28, 1100–1110 (2006); Member-Vidal, Enrique
6. Davis, J.V., Kulis, B., Jain, P., Sra, S., Dhillon, I.S.: Information-theoretic metric learning. In: ICML 2007, pp. 209–216 (2007)
7. Brighton, H., Mellish, C.: Advances in instance selection for instance-based learning algorithms. Data Mining and Knowledge Disc. 6, 153–172 (2002)
8. Zuo, W., Zhang, D., Wang, K.: On kernel difference-weighted k-nearest neighbor classification. Pattern Anal. Appl. 11, 247–257 (2008)
9. Holmes, C.C., Adams, N.M.: Likelihood inference in nearest-neighbour classification models. Biometrika 90, 99–112 (2003)

10. Marin, J.M., Robert, C.P., Titterington, D.M.: A Bayesian reassessment of nearest-neighbor classification. J. of the Am. Stat. Assoc. (2009)
11. Athitsos, V., Sclaroff, S.: Boosting nearest neighbor classi.ers for multiclass recognition. In: CVPR 2005: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005) - Workshops, vol. 45 (2005)
12. Schapire, R.E., Singer, Y.: Improved boosting algorithms using confidence-rated predictions. Machine Learning 37, 297–336 (1999)
13. Zou, H., Zhu, J., Hastie, T.: New multicategory boosting algorithms based on multicategory fisher-consistent losses. Annals of Applied Statistics 2(4), 1290–1306 (2008)
14. Nock, R., Nielsen, F.: Bregman divergences and surrogates for learning. IEEE Trans. PAMI 31, 2048–2059 (2009)
15. Bartlett, P., Jordan, M., McAuliffe, J.D.: Convexity, classification, and risk bounds. J. of the Am. Stat. Assoc. 101, 138–156 (2006)
16. Freund, Y., Schapire, R.E.: A Decision-Theoretic generalization of on-line learning and an application to Boosting. Journal of Comp. Syst. Sci. 55, 119–139 (1997)
17. Fei-Fei, L., Perona, P.: A bayesian hierarchical model for learning natural scene categories. In: CVPR, pp. 524–531 (2005)
18. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR, pp. 2169–2178 (2006)
19. 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)

Appendix

Proofsketch of Theorem 2. Without loss of generality and to simplify notations, assume that $j = t$ in Alg. 1, and denote $\boldsymbol{w}_j$ the weight vector on which we compute (12) — thus, the weight update in (13) gives $\boldsymbol{w}_{t+1}$, and the first weight vector is $\boldsymbol{w}_1$. Let us denote $Z_t \doteq ||\boldsymbol{w_{t+1}}||_1$ the normalization coefficient for weights, and $\tilde{w}_{(t+1)i} \doteq w_{ti}/Z_t$ the normalized weight of example $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ in the $(t+1)^{\text{th}}$ weight vector. Few derivations lead:

$$\varepsilon^{\exp}(\boldsymbol{h}^\ell, \mathcal{S}) = \prod_{t=1}^{T} Z_t \ . \tag{24}$$

We now compute an upperbound for Z_t, removing the t index for readability. For this objective, we extend notations (16) to the normalized tilda notation above, and let $\underline{\varrho} \doteq \min_{i,t:K(\boldsymbol{x}_i,\boldsymbol{x}_t)>0} K(\boldsymbol{x}_i, \boldsymbol{x}_t)$ and $\overline{\varrho} \doteq \max_{i,t} K(\boldsymbol{x}_i \boldsymbol{x}_t)$. Due to the lack of place, we make the proof in the simpler case where $\underline{\varrho} = \overline{\varrho} = 1$. We obtain:

$$\begin{aligned}
Z = \sum_{i=1}^{m} \tilde{w}_i \exp\left(-\delta_j \mathrm{r}_{ij}\right) &\leq \tilde{w}_j^+ \exp\left(-\underline{\varrho}\frac{C-1}{C}\log\left(\frac{(C-1)w_j^+}{w_j^-}\right)\right) + \\
&+(1-\tilde{w}_j^- - \tilde{w}_j^+) + \tilde{w}_j^- \exp\left(\frac{\overline{\varrho}}{C}\log\left(\frac{(C-1)w_j^+}{w_j^-}\right)\right) = \\
&= 1 - \tilde{w}_j^- - \tilde{w}_j^+ + \frac{C}{C-1}\left((C-1)\tilde{w}_j^+\right)^{\frac{1}{C}}\left(\tilde{w}_j^-\right)^{1-\frac{1}{C}} = \\
&= 1 - (\tilde{w}_j^- + \tilde{w}_j^+)\left[1 - \frac{C\left((C-1)\tilde{\tilde{w}}_j^+\right)^{\frac{1}{C}}\left(\tilde{\tilde{w}}_j^-\right)^{1-\frac{1}{C}}}{C-1}\right]
\end{aligned} \tag{25}$$

where we have used the shorthands $\tilde{\tilde{w}}_j^+ \doteq w_j^+/(w_j^+ + w_j^-)$ and $\tilde{\tilde{w}}_j^- \doteq w_j^-/(w_j^- + w_j^+)$. Using the **WIA** (23) and the fact that $1 - x \leq \exp(-x)$, we obtain from (25):

$$Z \leq \exp\left[-\eta\left(1 - f(\tilde{\tilde{w}}_j^+)\right)\right] \ , \tag{26}$$

$$f(x) \doteq \frac{C}{C-1}\left((C-1)x\right)^{\frac{1}{C}}(1-x)^{1-\frac{1}{C}} \ , x \in [0,1] \ .$$

$f(x)$ is concave on $[0,1]$ and admits a maximum in $x = 1/C$; Assuming the **WIA** (22), we get $|\tilde{\tilde{w}}_j^+ - \frac{1}{C}| \geq \gamma$. If $x \leq \frac{1}{C} - \gamma$, then $f(x) \leq g_-(\gamma)$, and if $x \geq \frac{1}{C} + \gamma$, then $f(x) \leq g_+(\gamma)$, with:

$$g_-(\gamma) \doteq (1 - C\gamma)^{\frac{1}{C}}\left(1 + \frac{C}{C-1}\gamma\right)^{1-\frac{1}{C}} \ ,$$

$$g_+(\gamma) \doteq (1 + C\gamma)^{\frac{1}{C}}\left(1 - \frac{C}{C-1}\gamma\right)^{1-\frac{1}{C}} \ .$$

But it can be shown that both $g_-(\gamma)$ and $g_+(\gamma)$ can be upperbounded by $g(\gamma) = 1 - C\gamma^2/(C-1)$, $\forall C \geq 2, \forall \gamma \in [0,1]$. Plugging the bound in (26), we obtain:

$$Z \leq \exp\left[-\eta\left(1 - g(\tilde{\tilde{w}}_j^+)\right)\right] = \exp\left[-\frac{C}{C-1}\eta\gamma^2\right] \ .$$

Finally, $Z \leq 1$ because $0 \leq f_C(x) \leq 1$ for any $x \in [0,1]$, and (24) yields $\varepsilon^{\exp}(\boldsymbol{h}^\ell, \mathcal{S}) \leq \exp(-C\eta\gamma^2\tau/(C-1))$. Using the fact that $\varepsilon^{0/1}(\boldsymbol{h}^\ell, \mathcal{S}) \leq \varepsilon^{\exp}(\boldsymbol{h}^\ell, \mathcal{S})$ yields the proof of Theorem 2.

Video Based Face Recognition Using Graph Matching

Gayathri Mahalingam and Chandra Kambhamettu

Video/Image Modeling and Synthesis (VIMS) Laboratory,
Department of Computer and Information Sciences
University of Delaware, Newark, DE, USA

Abstract. In this paper, we propose a novel graph based approach for still-to-video based face recognition, in which the temporal and spatial information of the face from each frame of the video is utilized. The spatial information is incorporated using a graph based face representation. The graphs contain information on the appearance and geometry of facial feature points and are labeled using the feature descriptors of the feature points. The temporal information is captured using an adaptive probabilistic appearance model. The recognition is performed in two stages where in the first stage a Maximum a Posteriori solution based on PCA is computed to prune the search space and select fewer candidates. A simple deterministic algorithm which exploits the topology of the graph is used for matching in the second stage. The experimental results on the UTD database and our dataset show that the adaptive matching and the graph based representation provides robust performance in recognition.

1 Introduction

Face recognition has long been an active area of research, and numerous algorithms have been proposed over the years. For more than a decade, active research work has been done on face recognition from still images or from videos of a scene [1]. A detailed survey of existing algorithms on video-based face recognition can be found in [2] and [3]. The face recognition algorithms developed during the past decades can be classified into two categories: holistic approaches and local feature based approaches. The major holistic approaches that were developed are Principal Component Analysis (PCA) [4], combined Principal Component Analysis and Linear Discriminant Analysis (PCA+LDA) [5], and Bayesian Intra-personal/Extra-personal Classifier (BIC) [6].

Chellappa *et al.* [7] proposed an approach in which a Bayesian classifier is used for capturing the temporal information from a video sequence and the posterior distribution is computed using sequential importance sampling. As for the local feature based approaches, Manjunath and Chellappa [8] proposed a feature based approach in which features are derived from the intensity data without assuming any knowledge of the face structure. Topological graphs are used to represent relations between features, and the faces are recognized by matching the graphs. Ersi and Zelek [9] proposed a feature based approach where in a statistical Local

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 82–94, 2011.

Feature Analysis (LFA) method is used to extract the feature points from a face image. Gabor histograms are generated using the feature points and are used to identify the face images by comparing the Gabor histograms using a similarity metric. Wiskott *et al.* [10] proposed a feature based graph representation of the face images for face recognition in still images. The face is represented as a graph with the features as the nodes and each feature described using a Gabor jet. The recognition is performed by matching graphs and finding the most similar ones. A similar framework was proposed by Ersi *et al.* [11] in which the graphs were generated by triangulating the feature points.

Most of these approaches focused on image-based face recognition applications. Various approaches to video-based face recognition have been studied in the past, in which both the training and test set are video sequences. Video-based face recognition has the advantage of using the temporal information from each frame of the video sequence. Zhou *et al.* [12] proposed a probabilistic approach in which the face motion is modeled as a joint distribution, whose marginal distribution is estimated and used for recognition. Li [13] used the temporal information to model the face from the video sequence as a surface in a subspace and performed recognition by matching the surfaces. Kim *et al.* [14] recognized faces from video sequences by fusing pose-discriminant and person-discriminant features by modeling a Hidden Markov Model (HMM) over the duration of a video sequence. Stallkamp *et al.* [15] proposed a classification sub-system of a real-time video-based face identification system. The system uses K-nearest neighbor model and Gaussian mixture model (GMM) for classification purposes and uses distance-to-model, and distance-to-second-closest metrics to weight the contribution of each individual frame to the overall classification decision.

Liu and Chen [16] proposed an adaptive HMM to model the face images in which the HMM is updated with the result of identification from the previous frame. Lee *et al.* [17] represented each individual by a low-dimensional appearance manifold in the ambient image space. The model is trained from a set of video sequences to extract a transition probability between various poses and across partial occlusions. Park and Jain [18] proposed a 3D model based approach in which a 3D model of the face is used to estimate the pose of the face in each frame and then matching is performed by extracting the frontal pose from the 3D model. Xu *et al.* [19] proposed a video based face recognition system in which they integrate the effects of pose and structure of the face and the illumination conditions for each frame in a video sequence in the presence of multiple point and extended light sources. The pose and illumination estimates in the probe and gallery sequences are then compared for recognition applications.

In this paper, we propose a novel graph based approach for image-to-video based face recognition which utilizes the spatial and temporal characteristics of the face from the videos. The face is spatially represented by constructing a graph using the facial feature points as vertices and labeling them with their feature descriptors. A probabilistic mixture model is constructed for each subject which captures the temporal information. The recognition is performed in two stages where in the first stage the probabilistic mixture model is used to prune

the search space using a MAP rule. A simple deterministic algorithm that uses cosine similarity measure is used to compare the graphs in the second stage. The probabilistic models are updated with the results of recognition from each frame of the video sequence, thus making them adaptive. Section 2 explains our procedure in constructing the graphs and the adaptive probabilistic mixture models for each subject. The two stage recognition is explained in section 5.

2 Face Image Representation

In this section, we describe our approach in extracting the facial feature points and their descriptors which are used in the spatial representation of the face images. Every face is distinguished not by the properties of individual features, but by the contextual relative location and comparative appearance of these features. Hence it is important to identify those features that are conceptually common in every face such as eye corners, nose, mouth, etc. In our approach, the facial feature points are extracted using a modified Local Feature Analysis (LFA) technique, and extracted feature points are described using Local Binary Pattern (LBP) [20], [21] feature descriptors.

2.1 Feature Point Extraction

The Local Feature Analysis (LFA) proposed by Penev and Atick [22] constructs kernels, which are basis vectors for feature extraction. The kernels are constructed using the eigenvectors of the covariance matrix of the vectorized face images. LFA is referred to as a local method since it constructs a set of kernels that detects local structure; e.g., nose, eye, jaw-line, and cheekbone, etc. The local kernels are optimally matched to the second-order statistics of the input ensemble [22]. Given a set of n d-dimensional images $x_1, \ldots, x_n$, Penev and Atick [22] compute the covariance matrix C, from the zero-mean matrix X of the n vectorized images as follows:

$$C = XX^T. \tag{1}$$

The eigenvalues of the covariance matrix C are computed and the first k largest eigenvalues, $\lambda_1, \lambda_2, \ldots, \lambda_k$, and their associated eigenvectors $\psi_1, \ldots, \psi_k$ to define the kernel K,

$$K = \Psi \Lambda \Psi^T \tag{2}$$

where $\Psi = [\psi_1 \ldots \psi_k]$, $\Lambda = diag(\frac{1}{\sqrt{\lambda_r}})$.

The rows of K contain the kernels. These kernels have spatially local properties and are "topographic" in the sense that the kernels are indexed by spatial location of the pixels in the image, *i.e.*, each pixel in the image is represented by a kernel from K. Figure 1(a) shows the kernels corresponding to the nose, eye, mouth and cheek positions. The kernel matrix K transforms the input image matrix X to the LFA output $O = K^T X$ which inherits the same topography as the input space.

Hence, the dimension of the output is reduced by choosing a subset of kernels, M, where M is a subset of indices of elements of K. These subsets of kernels are considered to be at those spatial locations which are the feature points of the face image. Penev and Atick [22] proposed an iterative algorithm that uses the mean reconstruction error to construct M by adding a kernel at each step whose output produces the maximum reconstruction error,

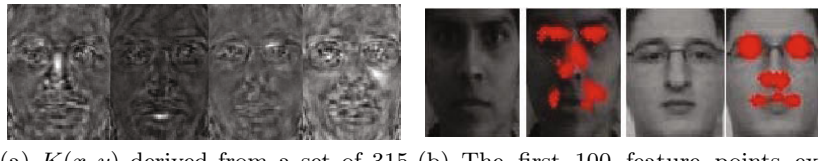
$$\arg\max_{x} \langle ||O(x) - O^{rec}(x)||^2 \rangle \tag{3}$$

where $O^{rec}(x)$ is the reconstruction of the output $O(x)$.

Although mean reconstruction error is a useful criterion for representing data, it does not guarantee an effective discrimination between data from different classes as the kernels selection process aims at reducing the reconstruction error for the entire image and not the face region. Hence, we propose to use the Fisher's linear discriminant method [23] to select the kernels that characterize the most discriminant and descriptive feature points of different classes. We compute the Fisher scores using the LFA output O. Fisher score is a measure of discriminant power which estimates how well different classes of data are separated from each other, and is measured as the ratio of variance between the classes to the variance within the classes. Given the LFA output $O = [o_1 \dots o_n]$ for c classes, with each class having n_i samples in the subset χ_i, the Fisher score of the x^{th} kernel, $J(x)$ is given by

$$J(x) = \frac{\sum_{i=1}^{c} n_i (m_i(x) - m(x))^2}{\sum_{i=1}^{c} \sum_{o \in \chi_i} (o(x) - m_i(x))^2} \tag{4}$$

where $m(x) = \frac{1}{n} \sum_{i=1}^{c} n_i m_i(x)$ and $m_i(x) = \frac{1}{n_i} \sum_{o \in \chi_i} o(x)$. The kernels that correspond to high Fisher scores are chosen to represent the most discriminative feature points of the image. Figure 1(b) shows the set of feature points extracted using the Fisher scores.



(a) $K(x, y)$ derived from a set of 315 images

(b) The first 100 feature points extracted from the training images

Fig. 1. 1(a) shows $K(x, y)$ at the nose, mouth, eye, and cheeks and 1(b) shows the feature points extracted (best viewed in color)

2.2 Feature Description with Local Binary Pattern

A feature descriptor is constructed for each feature point extracted from an image using Local Binary Pattern (LBP).

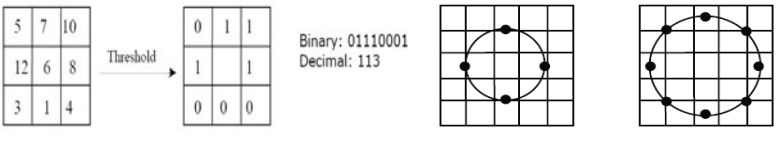
The original Local Binary Pattern (LBP) operator proposed by Ojala *et al.* [20] is a simple but very efficient and powerful operator for texture description. The operator labels the pixels of an image by thresholding the $n \times n$ neighborhood of each pixel with the value of the center pixel, and considering the result value as a binary number. Figure 2(a) shows an example of the basic LBP operator and figure 2(b) shows a $(4, 1)$ and $(8, 2)$ circular LBP operator. The histogram of the labels of the pixels of the image can be used as a texture descriptor. The grey-scale invariance is achieved by considering a local neighborhood for each pixel, and invariance with respect to scaling of the grey scale is achieved by considering just the signs of the differences in the pixel values instead of their exact values. The LBP operator with P sampling points on a circular neighborhood of radius R is given by,

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \tag{5}$$

where

$$s(x) = \begin{cases} 1 \text{ if } x \geq 0 \\ 0 \text{ if } x < 0 \end{cases} \tag{6}$$

Ojala *et al.* [21] also introduced another extension to the original operator which uses the property called *uniform patterns* according to which a LBP is called uniform if there exist at most two bitwise transitions from 0 to 1 or vice versa. Uniform patterns can reduce the dimension of the LBP significantly which is advantageous for face recognition.



(a) The Basic LBP operator (b) $(4, 1)$ and $(8, 2)$ circular LBP operator

Fig. 2. The basic LBP operator and the circular LBP operator

In our experiments, we use $LBP^{u2}_{8,2}$ operator which denotes a uniform LBP operator with 8 sampling pixels in a local neighborhood region of radius 2. A 5×5 window around the pixel is chosen as the neighborhood region and a feature vector of length 59 is obtained.

3 Image Graph Construction

The most distinctive property of a graph is its geometry, which is determined by the way the vertices of the graph are arranged spatially. Graph geometry plays

an important role in discriminating the graphs of different face images. In our approach, the graph geometry is defined by constructing a graph with constraints imposed on the length of the edges between a vertex and its neighbors.

Considering that we extract around n feature points from each face image, at least $n!$ graphs can be generated for each image. Evaluating this number of graphs for each probe image would be very computationally expensive. Hence, a graph generating procedure that generates a unique graph with the given set of vertices is proposed. At each iteration, vertices and edges are added to the graph in a Breadth-first search manner and considering a spatial neighborhood distance for each vertex. This generates a unique graph for a set of feature points. The procedure to generate a graph given a set of vertices is given as follows;

```
1 Pick a random vertex v from the list of vertices of the graph.
2 Add v to the end of the queue q.
3 While NOT all the vertices have been visited
      Pick a vertex u from the front of the queue q.
      If u is not visited
          Find the Neighbors N of u who are within a Euclidean distance.
          Add N to the queue q.
          Mark u as visited
      endif
  endwhile
```

The idea behind representing face images using graphs is mainly due to the spatial properties of the graph, as a graph can represent the inherent shape changes of a face and also provide a simple, but powerful matching technique to compare graphs.

4 Probabilistic Graph Appearance Model

The appearance of a graph is another important distinctive property and is described using the feature descriptors of the vertices of the graph. An efficient as well as effective description of the appearance of the vertices of the graphs is required in order construct a graph appearance model that elevates the distinctive properties of the face of an individual. Modeling the joint probability distribution of the appearance of the vertices of the graphs of an individual produces an effective representation of the appearance model through a probabilistic framework. Since the model is constructed using the feature descriptors, it is easy to adapt the model to the changes in the size of the training data for the individual. Given N individuals and M training face images, the algorithm to learn the model is described as follows:

1. Initialize N model sets.
2. For each training image I_c^j, (j^{th} image of the c^{th} individual)
   a. Extract the feature points (as described in Subsection 2.1).

b. Compute feature descriptors for each feature point (as described in Subsection 2.2).
c. Construct Image graphs (as described in Subsection 3).
d. Include the graph in the model of the c^{th} individual.

3. Construct the appearance model for each individual using their model sets.

In our approach, a probabilistic graph appearance model is generated for each subject and is used for training purposes. Given a graph $G(V, E)$, where V is the list of vertices in the graph, and E the set of edges in the graph, the probability of G belonging to a model set (subject) k is given by,

$$R_k = \max_n P(G|\Phi_n) \tag{7}$$

where $P(G|\Phi_n)$ is the posterior probability, and Φ_n is the appearance model for the n^{th} subject constructed using the set of feature descriptors F of the set of vertices of all the graphs of the subject. The appearance model Φ_n is constructed by estimating the joint probability distribution of the appearance of the graphs for each subject. R_k is called the Maximum a Posteriori (MAP) solution. In our approach, we estimate the joint probability distribution of the graph appearance model for each subject using the Gaussian Mixture Model (GMM) [24] which can efficiently represent heterogeneous data, the dominant patterns which are captured by the Gaussian component distributions.

Given a training face database containing images of L subjects and each subject having at least one image in the training database, the set of feature descriptors X for each subject to be used to model the joint likelihood of the subject will be a $(m \times f) \times t$ distribution, where m is the number of images for each subject, f is the number of feature points extracted for each image and t the dimension of the feature vector (in our case, it is 59 and is reduced to 20). To make the appearance model estimation more accurate and tractable, we use the Principal Component Analysis (PCA) to reduce the dimensionality of the feature vectors.

Each subject in the database is modeled as a GMM with K Gaussian components. The set of feature descriptors X of each subject is used to model the GMM of that individual. Mathematically, a GMM is defined as:

$$P(X|\theta) = \sum_{i=1}^{K} w_i.N(X|\mu_i, \sigma_i). \tag{8}$$

where

$$N(X|\mu_i, \sigma_i) = \frac{1}{\sigma_i\sqrt{2\pi}}.e^{-\frac{(X-\mu_i)^2}{2\sigma^2}}$$

are the components of the mixture, $\theta = \{w_i, \mu_i, \sigma_i^2\}_{i=1}^K$ includes the parameters of the model, which includes the weights w_i, the means μ_i, and the variances σ_i^2 of the K Gaussian components.

In order to maximize the likelihood function $P(X|\theta)$, the model parameters are re-estimated using the Expectation-Maximization (EM) technique [25].

The EM algorithm is an iterative procedure to compute the Maximum Likelihood (ML) estimate in the presence of missing or hidden data. In ML estimation, we wish to estimate the model parameters for which the observed data are the most likely:

$$\theta^c = arg \max_{\theta} P(X|\theta). \tag{9}$$

At each iteration of the EM algorithm the missing data are estimated with the current estimate of the model parameters, and the likelihood function is maximized with assumption that the missing data are known. For more details about the EM algorithm see [25].

5 Adaptive Matching and Recognition

In this section, we describe our two stage matching procedure to adaptively match every frame of the video sequence and the trained appearance models and the graphs. In the first stage of the matching process, a MAP solution is computed for the test graph using the trained appearance models. The MAP solution is used to prune the search space for the second stage of matching. A subset of individuals' appearance model and their trained graphs are selected based on the MAP solution. This subset of appearance models are used in the second stage of matching process. In the second stage, a simple deterministic algorithm that uses the cosine similarity measure and the nearest neighborhood classifier to find the geometrical similarity of the graphs is proposed. The GMM is adapted with the result of recognition from each frame of the test video sequence. We use the likelihood score and the graph similarity score to decide on the correctness of the recognition and update the appropriate GMM. The recognition result of a frame is considered correct if the difference between the highest likelihood score and the second highest likelihood score is greater than a threshold. A similar difference in graph similarity scores is also computed to support the decision. This measure of correctness is based on the same idea as Lowe [26], that reliable matching requires the best match to be significantly better than the second-best match. For a given test sequence, the difference in the likelihood scores and the difference in the similarity scores are computed and the GMM is updated if these values are greater than a threshold. Given an existing GMM Θ_{old} and observation vectors O from the test sequence, the new GMM is estimated using the EM algorithm with Θ_{old} as the initial values. The entire matching procedure is given as follows;

1. For *each frame f in the video sequence*
   a. Extract the facial feature points and their descriptors from *f*.
   b. Reduce the dimension of feature descriptors using the projection matrix from training stage.
   c. Construct the image graph G.
   d. Obtain the probability of G belonging to each appearance model, and select the k model sets with highest probability. k is 10% in our experiments.

e. Obtain the similarity scores between G and the graphs of k individuals.
f. Update the appropriate appearance model based on the likelihood score and similarity score.

2. Select the individual with the maximum number of votes from all the frames.

The algorithm to find the spatial similarity between two graphs is given as follows;

1. For each vertex v in the test graph with a spatial neighborhood W, a search is conducted over W (in the trained graph) and the best matching feature vertex u is selected, such that

$$S_{vu} = \frac{f_v . f_u}{|f_v||f_u|} \tag{10}$$

where f_v and f_u are the feature vectors of v and u respectively, and S_{vu} is the similarity score between v and u.

2. Repeat step 1 with neighbors of v and so on until all the vertices have been matched. The sum of the similarity scores of all the vertices gives the measure of similarity between the two graphs.

6 Experiments

In order to validate the robustness of the proposed technique, we used a set of close range and moderate range videos from the UTD database [27]. The database included 315 subjects with high resolution images in various poses. The videos included subjects with neutral expression and also walking towards the camera from a distance. We also generated a set of moderate range videos (both indoor and outdoor) with 6 subjects. Figure 3 shows sample video frames from the UTD dataset and figure 4 shows sample video frames from our dataset.

In the preprocessing step, the face region is extracted from the image, normalized using histogram equalization technique and are resized to 72×60 pixels. 150 features were extracted and a LBP is computed for each feature point. PCA is performed on the feature vectors to reduce the dimension from 59 to 20 (with nearly 80% of the non-zero eigenvalues retained). A graph is generated for each face image with a maximum spatial neighborhood distance of 30 pixels. A graph space model is constructed for each subject using GMM with 10 Gaussian components.

During the testing stage, in order to mimic the practical situation, we consider a subset of frames in which an individual appear in the video and use it for testing purposes. We randomly select an individual and a set of frames that include the individual. The preprocessing and the graph generation procedure similar to those performed in the training stage are applied to each frame of the video sequence. The likelihood scores are computed for the test graph and the GMMs and the training graphs are matched with the test graph to produce similarity scores, and the appropriate GMM is updated using the similarity and likelihood

(a) Sample frames from close-range videos of UTD dataset

(b) Sample frames from moderate-range videos of UTD dataset

Fig. 3. Sample video frames from the UTD video dataset

Table 1. Comparison of the error rates with different algorithms

	HMM	AGMM	Graphs	AGMM+Graphs
UTD Database (close-range)	24.3%	24.1%	23.2%	20.1%
UTD Database (moderate-range)	31.2%	31.2%	29.8%	25.4%
Our Dataset	8.2%	3.4%	2.1%	1.1%

scores. The threshold is determined by the average of the difference in likelihood scores and similarity scores between each class of data. Though the threshold value is data dependent, the average proves to be an optimum value.

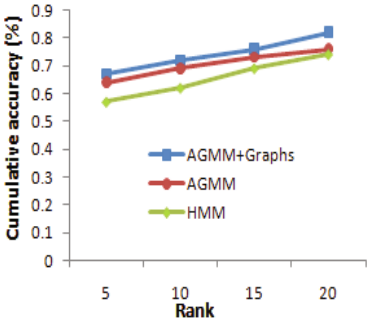
The performance of the algorithm is compared with video-based recognition algorithm in [16] which handles video-to-video based recognition. The algorithm in [16] performs eigen analysis on the face images and uses an adaptive Hidden Markov Model (HMM) for recognition. We also test the performance of the system with only the adaptive graph appearance model (AGMM) and the appearance model with the graph model sets (AGMM+Graphs). The results are tabulated in the Table 1. Figure 5 shows the Cumulative Match Characteristic curve obtained for various algorithms (HMM, AGMM and AGMM+Graphs) on the UTD dataset.

From the error rates we can see that the performance of our approach is definitely promising when compared with the other approaches. The account of spatial and temporal information together improves the performance of the recognition process. The number of images in the training dataset played an important role in the performance, as it is evident from the error rates. The close-range videos of the UTD database has lower error rates than the moderate-range videos. This is due to the reason that the frame of the video sequence mostly contains the face region thus gathering more details of the facial features than the moderate-range videos. The number of training set images for each subject played a role in the performance. The UTD dataset included 3 training images

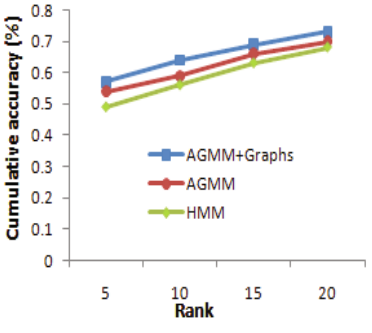
(a) Sample frames from indoor videos of our dataset

(b) Sample frames from outdoor videos of our dataset

Fig. 4. Sample video frames from our video dataset



(a) CMC curve for close-range videos of UTD database



(b) CMC curve for moderate-range videos of UTD database

Fig. 5. Cumulative Match Characteristic curves for close-range and moderate-range videos

for each subject whereas our dataset included at least 5 training images. The algorithm shows a high recognition rate when experimented on our dataset as it can be seen for the error rates. Though there were limited number of subjects in the dataset, the videos in the dataset included both indoor and outdoor videos taken using a PTZ camera which is mainly used for surveillance. The system provided a better performance with both indoor and outdoor videos which has different illumination, pose changes and in moderate range.

The system performs better as a video based face recognition system than a still image based face recognition system, due to the wealth of temporal information available from the video sequence and the effective use of it by the proposed adaptive probabilistic model. As a still image based face recognition, an image with a frontal pose of the face yields better performance than non frontal pose image. Thus, pose of the face image plays a role in the recognition. Also, the system's performance is affected by the comparison of a single high resolution image with a low resolution frame in a still image based face recognition system. Thus

the adaptive matching technique combined with the graph based representation is significantly an advantage in matching images with videos.

From our experiments, we found that changing the value of the parameters did not significantly change the performance of the system and the values that we used tend to be the optimum. For example, increasing the maximum Euclidean distance between two vertices of a graph to a value greater than the width or length of the image will have no effect as the graph will always be connected as the distance between two vertices will never be greater than these values.

7 Conclusion

In this paper, we proposed a novel technique for face recognition from videos. The proposed technique utilizes both the temporal and spatial characteristics of a face image from the video sequence. The temporal characteristics are captured by constructing a probabilistic appearance model and a graph is constructed for each face image using the set of feature points as vertices of the graph and labeling it with the feature descriptors. A modified LFA and LBP were used to extract the feature points and feature descriptors respectively. The appearance model is built using GMM for each individual in the training stage and is adapted with the recognition results of each frame in the testing stage. A two stage matching procedure that exploits the spatial and temporal characteristics of the face image sequence is proposed for efficient matching. A simple deterministic algorithm to find similarity between the graphs is also proposed. Our future work will handle video sequences involving various pose of the faces, different resolutions, and video-to-video based recognition.

References

1. Chellappa, R., Wilson, C.L., Sirohey, S.: Human and machine recognition of faces: a survey. Proceedings of the IEEE 83, 705–741 (1995)
2. Wang, H., Wang, Y., Cao, Y.: Video-based face recognition: A survey. World Academy of Science, Engineering and Technology 60 (2009)
3. Zhao, W., Chellappa, R., Phillips, P.J., Rosenfeld, A.: Face recognition: A literature survey (2000)
4. Turk, M., Pentland, A.: Eigenfaces for recognition. Journal of Cognitive Neuroscience 3, 71–86 (1991)
5. Etemad, K., Chellappa, R.: Discriminant analysis for recognition of human face images. Journal of the Optical Society of America 14, 1724–1733 (1997)
6. Moghaddam, B., Nastar, C., Pentland, A.: Bayesian face recognition using deformable intensity surfaces. In: Proceedings of Computer Vision and Pattern Recognition, pp. 638–645 (1996)
7. Zhou, S., Krueger, V., Chellappa, R.: Probabilistic recognition of human faces from video. In: Computer Vision and Image Understanding, vol. 91, pp. 214–245 (2003)
8. Manjunath, B.S., Chellappa, R., Malsburg, C.: A feature based approach to face recognition (1992)
9. Ersi, E.F., Zelek, J.S.: Local feature matching for face recognition. In: Proceedings of the 3rd Canadian Conference on Computer and Robot Vision (2006)

10. Wiskott, L., Fellous, J.M., Kruger, N., Malsburg, C.V.D.: Face recognition by elastic bunch graph matching. IEEE Trans. on Pattern Analysis and Machine Intelligence 19, 775–779 (1997)
11. Ersi, E.F., Zelek, J.S., Tsotsos, J.K.: Robust face recognition through local graph matching. Journal of Multimedia, 31–37 (2007)
12. Zhou, S., Krueger, V., Chellappa, R.: Face recognition from video: A condensation approach. In: Proc. of Fifth IEEE Internation Conference on Automatic Face and Gesture Recognition, pp. 221–228 (2002)
13. Li, Y.: Dynamic face models: construction and applications. Ph.D. Thesis, University of London (2001)
14. Kim, M., Kumar, S., Pavlovic, V., Rowley, H.A.: Face tracking and recognition with visual constraints in real-world videos. In: CVPR (2008)
15. Stallkamp, J., Ekenel, H.K.: Video-based face recognition on real-world data (2007)
16. Liu, X., Chen, T.: Video-based face recognition using adaptive hidden markov models. In: CVPR (2003)
17. Lee, K.C., Ho, J., Yang, M.H., Kriegman, D.: Visual tracking and recognition using probabilistic appearance manifolds. Computer Vision and Image Understanding 99, 303–331 (2005)
18. Park, U., Jain, A.K.: 3D model-based face recognition in video. In: Lee, S.-W., Li, S.Z. (eds.) ICB 2007. LNCS, vol. 4642, pp. 1085–1094. Springer, Heidelberg (2007)
19. Wu, Y., Roy-Chowdhury, A., Patel, K.: Integrating illumination, motion and shape models for robust face recognition in video. EURASIP Journal of Advances in Signal Processing: Advanced Signal Processing and Pattern Recognition Methods for Biometrics (2008)
20. Ojala, T., Pietikainen, M., Harwood, D.: A comparative study of texture measures with classification based on feature distributions. Pattern Recognition, 51–59 (1996)
21. Ojala, T., Pietikainen, M., Maenpaa, T.: A generalized local binary pattern operator for multiresolution gray scale and rotation invariant texture classification. In: Second International Conference on Advances in Pattern Recognition, Rio de Janeiro, Brazil, pp. 397–406 (2001)
22. Penev, P., Atick, J.: Local feature analysis: A general statistical theory for object representation. Network: Computation in Neural Systems 7, 477–500 (1996)
23. Mika, S., Rätsch, G., Weston, J., Schölkopf, B., Müller, K.R.: Fisher discriminant analysis with kernels (1999)
24. McLachlam, J., Peel, D.: Finite mixture models (2000)
25. Redner, R.A., Walker, H.F.: Mixture densities, maximum likelihood and the em algorithm. SIAM Review 26, 195–239 (1984)
26. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60, 91–110 (2004)
27. O'Toole, A., Harms, J., Hurst, S.L., Pappas, S.R., Abdi, H.: A video database of moving faces and people. IEEE Transactions on Pattern Analysis and Machine Intelligence 27, 812–816 (2005)

A Hybrid Supervised-Unsupervised Vocabulary Generation Algorithm for Visual Concept Recognition

Alexander Binder[1], Wojciech Wojcikiewicz[1,2], Christina Müller[1,2], and Motoaki Kawanabe[1,2]

[1] Berlin Institute of Technology, Machine Learning Group, Franklinstr. 28/29, 10587 Berlin, Germany
{alexander.binder@,wojwoj@mail.}tu-berlin.de
[2] Fraunhofer Institute FIRST, Kekuléstr. 7, 12489 Berlin, Germany
{motoaki.kawanabe,christina.mueller}@first.fraunhofer.de

Abstract. Vocabulary generation is the essential step in the bag-of-words image representation for visual concept recognition, because its quality affects classification performance substantially. In this paper, we propose a hybrid method for visual word generation which combines unsupervised density-based clustering with the discriminative power of fast support vector machines. We aim at three goals: breaking the vocabulary generation algorithm up into two sections, with one highly parallelizable part, reducing computation times for bag of words features and keeping concept recognition performance at levels comparable to vanilla k-means clustering. On the two recent data sets Pascal VOC2009 and ImageCLEF2010 PhotoAnnotation, our proposed method either outperforms various baseline algorithms for visual word generation with almost same computation time or reduces training/test time with on par classification performance.

1 Introduction

Bag of words features [1] have turned into a widely-acknowledged tool for concept recognition which has shown superior performance in many recent contests on wide-domain image collections with high background and concept variance as well as presence of clutter [2,3,4,5]. Most prominent methods for visual vocabulary generation are unsupervised techniques like the density-based k-means algorithm, radius based clustering [6], and supervised methods like extremely randomized clustering forests (ERCF) [7,8,9]. Since people have tried more and more difficult images recently, one can observe an increase in the typical word size. See for example the 300 words used in [10] on the seminal Caltech101 benchmark [11] which has less clutter and rather low variance versus 4000 words [12] on Pascal VOC challenge data. Such an increase implies higher running times during visual vocabulary creation and bag of word computation. Several schemes have been proposed to deal with the runtime issue like hierarchical clustering [13,14] or ERCF. From our own experience both methods can suffer

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 95–108, 2011.

drawbacks in concept recognition performance compared to k-means clustering even though hierarchical k-means (HKM) speeds up notably over k-means and ERCF shows superb computational efficiency. In this paper, we propose a novel algorithm which uses the hierarchical clustering idea together with linear support vector machines (SVM) trained locally within each cluster and and has faster computation speeds in theory compared to vanilla k-means-based bag of words representations, while still maintaining the recognition performance of k-means visual vocabularies.

This paper is organized as follows. In Section 2, we explain our hybrid combination approach. After describing the datasets in Section 3 and the experimental setup in Section 4, we compare our method in Section 5 against k-means, hierarchical k-means and ERCF baselines on the two recent datasets Pascal VOC2009 and ImageCLEF2010 PhotoAnnotation.

2 Visual Word Generation

Bag of word features are based on three steps. At first one computes for each image a set of base features. In the second step the base features from the training data are used for computing a discretization of the input space of the base features into N regions. In the third step the base features extracted from one image are used to compute a histogram of dimensionality N based on assignments of the base features to the bins of the discretization obtained in the second step.

2.1 Hybrid Supervised-Unsupervised Approach

In order to generate a vocabulary of N visual words, we start with an unsupervised clustering of the base features into $N/2$ centers. For simplicity we relied on k-means clustering. At each of the clusters we train one support vector machine (SVM) in order to divide the cluster region into two parts. This gives rise to a partition of the space of the base features into N regions. The binary labels of the base features used for the SVM training are constructed from the image labelling which is inherited down to the base features belonging to one image, as we will explain in the following.

For multi-label concept recognition problems one issue remains to be solved. Each image can belong to several concept classes. At each cluster we have to select one partition of the set of all concept labels into two sets used for labelling the base features for binary SVM training. Since we are not interested in a perfect classification at a local cluster, we adopted an approximate randomized process to obtain good candidates for partitioning of the label set to be used to define a binary labeling.

The candidate labelling generation process was motivated by two ideas. We want to select a binarization such that

1. **balancedness:** the number of base features having a label in the positive class is approximately half of all base features within one cluster.

2. **overlap:** the number of base features which have at the same time labels in the positive as well as the negative class is low.

The second constraint comes from the fact that a base feature is assigned for SVM training to the positive class if it has at least one image label in the set of positive classes. We assign all image labels to the base features independent of the position of the base features within an image. This can lead to an adversary situation where a base feature is assigned to the positive class although its position in the image belongs to an object from the negative class. Such a problem can be avoided in an object detection scenario with bounding box or object position information which is not available here.

We implemented the first constraint as follows. At first, starting from an empty set for positive classes, we randomly draw a class from the set of all concepts and add to the positive set. Then, we iterate this procedure, i.e. we add a class selected randomly from the set of the remaining categories. This gives a series of growing sets of positive classes. For each of these sets within the series we can count how many of the base features will be labeled positive, because they have at least one label belonging to the set of positive classes. Let S be the set of all base features and $l(S)$ be its original multi-label vector, then we count

$$bal(positives) := \left| \, |S|/2 - |\{s \in S \mid l(s) \cap positives \neq \emptyset\}| \, \right|, \tag{1}$$

where $|S|$ denotes the cardinality of the set S, i.e. the number of its elements. We select the set from this series which has the number of base features without the labels in the positive sets being closest to half of the total number of base features.

This procedure can be repeated M times to obtain M candidates which get subsequently checked for their overlap constraint. For the overlap constraint we count directly

$$ol := \{s \in S \mid l(s) \cap positives \neq \emptyset,\ l(s) \cap negatives \neq \emptyset\}, \tag{2}$$

which is the number of the base features labeled with concepts in positive and negative sets simultaneously and select the best T candidates to be used to define a binary labelling for SVM training.

Each of the T candidates was evaluated using five-fold cross validation in order to select one final classifier used at a local cluster node. This is admittably inspired by the ERCF algorithm which also uses a randomized generation of candidate partitions. On the other hand, ERCF chooses local dichotomies along one axis and deploys an entropy criterion. The pseudo-code for the two-class labeling procedure is summarized at the next page.

2.2 Relation to the Baseline Procedures

Figure 1 illustrates the proposed method in comparison with the three baselines with a synthetic example of two class data marked with red and blue.

Generation of Candidates for Two-Class Labeling for SVM Training

```
choose M = number of random trials,
       T = number of candidates for SVM training
input: S = set of base features, l(S) their multi-labels
input: num_concepts= number of concepts in multi-label problem
output: Candidates = T partitions of the set of all concepts into two

for m=1:M
  positives(m,0) = {}

  for index=1:num_concepts
    class = random_select( concepts\positives(m,index-1) )
    positives(m,index) = {positives(m,index-1),class}
  end for

  i = argmin_{index} bal( positives(m,index) )
  mid(m) = positives(m,i)
  compute ol( mid(m) )
end for
Candidates = the T elements from mid with smallest overlap ol
```

The proposed procedure is comparable to hierarchical k-means (HKM) with $N/2$ clusters at the top level and 2 clusters at the second level. Because we deploy a supervised technique instead of k-means with 2 centers at each cluster, the proposed method can capture the class information correctly. On the other hand, k-means and HKM fail to separate the red and the blue classes. ERCF uses image labels as well, however its appealing speed gains come from restriction of the partition process to axis-parallel splits of the input space. The class boundary in Figure 1 is however not aligned to the axes, thus the proposed procedure works best in Figure 1.

Compared to vanilla k-means we have practical speed-ups of visual vocabulary generation with the proposed method, because our method requires a smaller number of clusters in the initial step. The local classifier training step consists of $N/2$ independent jobs which can be run separately on a vanilla CPU cluster, thus computation time of the extra step is negligible.

Another advantage comes at the step of computing the bag of words histograms: k-means requires for each base feature to compute N distances. This can however be speeded up empirically by computing a distance matrix between cluster centers and employing the triangle inequality to exclude certain candidate centers. The proposed method uses $N/2$ distance computations and one SVM function evaluation. For the linear SVMs we used here this amounts to computing one inner product. Note that for normalized base features $\|f\|^2 = const$ the distance computation is equivalent to an inner product

$$\|f - g\|^2 = 2c - 2\langle f, g\rangle \tag{3}$$

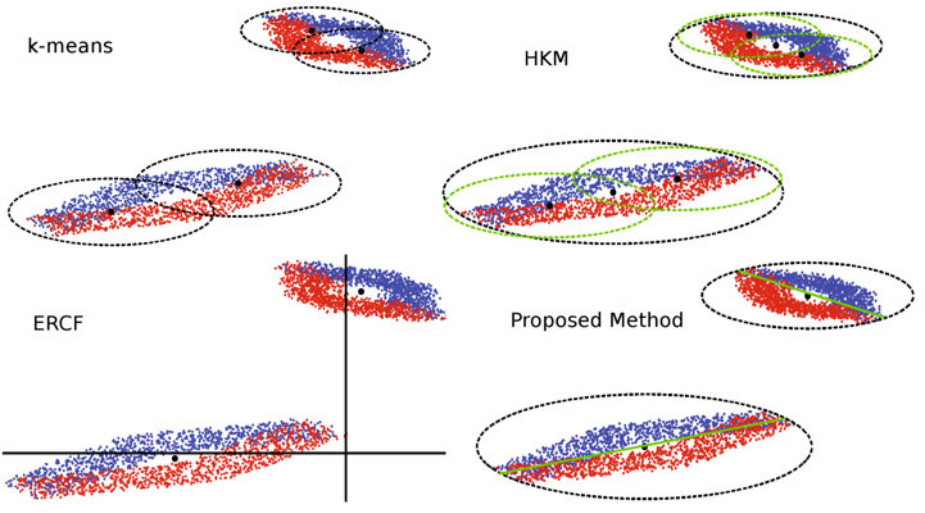


Fig. 1. An example of two blobs of two class data, marked red and blue and the results of different clustering algorithms with 4 clusters: Upper Left: k-means, Upper Right: Hierarchical k-means, Lower Left: ERCF, Lower Right: The proposed method

Thus, our method requires a computational amount of $N/2 + 1$ inner products compared to N inner products for vanilla k-means.

The advantage in computational speed enables us to double the word sizes of the standard k-means vocabularies with tiny extra runtime. Although the proposed method requires the same amount of time for feature computation as a k-means-based visual vocabulary, it increases the time to constructing kernels by a factor of two. We remark that this is still acceptable, because vocabulary generation is the bottle-neck in the entire process.

We have pursued a hybrid algorithm for visual word generation, where an unsupervised clustering method is done prior to the local supervised classification. This is because we do not expect to find a reasonable linear separation on the global input set of all base features. We conjectured that pure supervised procedures on entire base features such as ERCF can potentially suffer from degraded performance due to this problem. Furthermore, they also may have computational difficulties. The large cardinality in the order of hundred thousands or millions of input features necessary for visual word generation algorithms slows down linear SVMs and is still prohibitive for non-linear SVMs.

In this paper, we presented one special instance of an interpolation between unsupervised clustering and local classification. In general one can generate N visual words by using unsupervised clustering to obtain $N \cdot 2^{-k}$ base clusters and train k supervised classifiers at each cell which generates 2^k additional words at the $N \cdot 2^{-k}$ clusters.

Fig. 2. Pascal VOC2009 example images

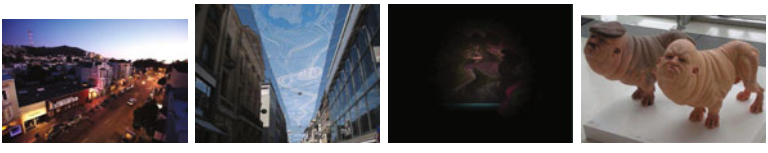


Fig. 3. ImageCLEF2010 example images

3 Datasets

Pascal VOC2009 data set has been used for the Pascal Visual Object Classification Challenge [4]. The part with disclosed labels is comprised of 7054 images falling into twenty object classes. The objects typically have highly varying sizes, positions and backgrounds.

ImageCLEF2010 PhotoAnnotation data set has in its labeled part 8000 images from flickr with 93 concept classes with highly variable concepts containing well defined objects as well as many rather ambiguously defined concepts like Citylife, Cute or Visual_Arts which makes it highly challenging for any recognition system.

4 Experimental Setup

As the base features, we computed for each image grey and rgb SIFT features [15] on a dense grid of pitch size six. Our choice is a reduced set inspired by the winners' procedures of ImageCLEF2009 PhotoAnnotation and Pascal VOC2009 [4]. The base features are clustered with the proposed method, and three baselines, i.e. vanilla k-means, hierarchical k-means (HKM) and ERCF. HKM extracts $N/2$ clusters at the top and 2 clusters at the lowest level which is close to the proposed method. Due to huge number of all base features, we used randomly drawn SIFT features for clustering: 2.4 millions for the grey color channel and 800000 for rgb-SIFT which has the triple dimensionality of grey SIFT. We have chosen these two color channels exemplarily for their different dimensionality as

it is known that the quality of density estimation which is implicitly performed by k-means may deteriorate in higher dimensions.

The local SVMs were trained with five-fold cross validation over regularization constants 0.25, 1 and 4. For each cluster, the best SVM was selected from 10 candidates generated by our procedure based on the cross validation scores. Due to computational costs, we limited for SVM training the number of base features to 5000 by random selection from all base features in each local cluster.

For visual concept classification, we deployed the χ^2-kernel based on bag-of-word features whose width is set to the average χ^2-distance. Then, all kernels were normalized to standard deviation one in Hilbert space, which allows to use the regularization constant 1 as a good approximative rule of thumb for SVM training, where we employ the shogun toolbox [16]. The performance is measured with average precision (AP) and area under curve (AUC) which are both threshold-invariant ranking measures. We evaluate all settings with 10 random splits to see statistical significance.

In order to advantages of proposed method in performance and runtime, we considered the following two settings.

Experiment 1. Comparison between the vocabularies with the same size of 500 words.

Experiment 2. Comparison of vanilla k-means with 4000 words and the other vocabularies with 8000 words.

In the first case, the vocabularies except for k-means can be computed much faster. Therefore, it is still OK, if the alternatives perform at least on par with k-means. In the second case, the larger 8000 vocabularies can be generated easily from the 4000 k-means prototypes by our algorithm and HKM with small computational costs, while ERCF can construct 8000 words quickly. Here, we are interested in performance gains over the standard bag-of-words procedure based on k-means.

5 Results

5.1 Concept Recognition Performance

The recognition performances using rgb SIFTs are summarized in Table 1 and 2 for VOC2009, and 3 and 4 for ImageCLEF2010. Results using grey SIFTs with qualitatively the same outcome are given in Appendix.

In Experiment 1, we compared vocabularies with 500 word in total. For both data sets (Table 1 & 3, the proposed method achieved slightly higher scores than k-means within shorter runtime, while the other faster variants degraded their performances to some extent. We further tried variants of our method with early-stopped clusters, where we reduced the number of k-means iterations to five. The performance did not drop much, because the clustering served only as a rough initialization for local classifications. This suggests that an exact density based clustering does not play a too large role and sheds an interesting

Table 1. Recognition performances of the baselines with 500 words versus those by our approach with 250 clusters and local SVMs (Experiment 1) on VOC2009 (summary from 10 repetitions)

Method / Score	AP	AUC
baseline: rgb-SIFT, hierarch KM250x2	43.87 ± 5.15	85.83 ± 1.71
baseline: rgb-SIFT, ERCF4x128	42.19 ± 5.33	85.39 ± 1.53
baseline: rgb-SIFT, KM500	44.46 ± 5.58	86.14 ± 1.78
proposed: rgb-SIFT, KM250, 5 iters+250 lin SVM	**45.16** ± 5.28	86.19 ± 1.70
proposed: rgb-SIFT, KM250+250 lin SVM	44.99 ± 5.25	**86.50** ± 1.45

Table 2. Recognition performances of the baselines with 4000/8000 words versus those by our approach with 4000 clusters and local SVMs (Experiment 2) on VOC2009 (summary from 10 repetitions)

Method / Score	AP	AUC
baseline: rgb-SIFT, hierarch KM4000x2	50.04 ± 5.18	88.04 ± 1.62
baseline: rgb-SIFT, ERCF16x512	47.54 ± 5.11	87.32 ± 1.54
baseline: rgb-SIFT, KM4000	48.94 ± 5.08	87.54 ± 1.73
proposed: rgb-SIFT, KM4000+4000 lin SVM	**52.70** ± 5.41	**89.11** ± 1.64

light on claims that density-based clustering is inferior to alternatives such as radius-based clustering [6,17].

In Experiment 2, we compared vocabularies whose sizes are closer to the ones used in recent competitions (k-means with 4000 words and the faster methods with 8000 words). Note that we did not compute vanilla k-means 8000 as clustering would take almost two weeks and is deemed too slow given the used setting. This is another way of fair comparisons, i.e. under equal time limitations. For VOC2009, our approach achieved notable gain over the k-means baseline, while HKM improved only slightly or ERCF even lost against the baseline. On the other hand, on ImageCLEF2010, all larger vocabularies of size 8000 did not

Table 3. Recognition performances of the baselines with 500 words versus those by our approach with 250 clusters and local SVMs (Experiment 1) on ImageCLEF2010 (summary from 10 repetitions)

Method / Score	AP	AUC
baseline: rgb-SIFT, hierarch KM250x2	32.53 ± 0.86	73.55 ± 1.30
baseline: rgb-SIFT, ERCF4x128	32.50 ± 1.40	73.62 ± 1.56
baseline: rgb-SIFT, KM500	33.07 ± 1.03	73.91 ± 1.44
proposed: rgb-SIFT, KM250, 5 iters+250 lin SVM	33.45 ± 0.94	74.38 ± 1.51
proposed: rgb-SIFT, KM250+250 lin SVM	**33.58** ± 0.92	**74.40** ± 1.41

Table 4. Recognition performances of the baselines with 4000/8000 words versus those by our approach with 4000 clusters and local SVMs (Experiment 2) on ImageCLEF2010 (summary from 10 repetitions)

Method / Score	AP	AUC
baseline: rgb-SIFT, hierarch KM4000x2,	36.29 ± 1.28	76.20 ± 1.50
baseline: rgb-SIFT, ERCF16x512	36.48 ± 1.19	76.04 ± 1.48
baseline: rgb-SIFT, KM4000,	36.16 ± 1.18	75.97 ± 1.40
proposed: rgb-SIFT, KM4000+4000 lin SVM	**36.78** ± 1.19	**76.60** ± 1.44

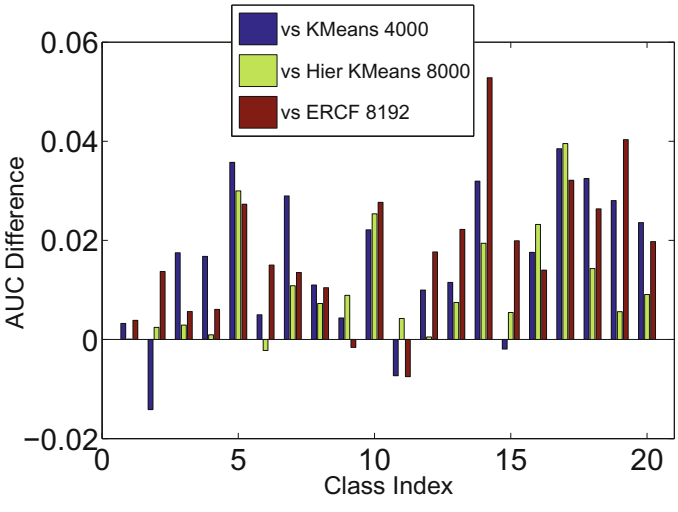


Fig. 4. Class-wise differences by AUC for VOC2009, rgb channel between proposed method, 8000 words and vanilla k-means 4000 words (left), hierarchical k-means 8000 words (mid), ERCF 8192 words (right)

improve the 4000 k-means significantly. We will see that for some of the abstract concepts in ImageCLEF2010 our algorithm degraded recognition performances.

By inspecting the differences between the proposed method and vanilla k-means in Figure 4 (left) and 5, we see some gains on most classes and a small fraction of setbacks For VOC2009 data we observe larger gains for classes bottle(5), cow(10) and pottedplant(16) which belonged to the rather difficult classes according to their performance on test data results for the winners' submission. For ImageCLEF2010 data the trends are more diverse. For the rgb channel we lose performance with the proposed method in 15 concepts out of 93 like birthday(65),grafitti(67),abstract(72),cat(76) and bicycle(82), while having many gains across a variety of narrow and broad concepts like in Partylife(1), River(14), Motion_Blur(39), Architecture(53), Visual_Arts(66), Train(84), Skateboard(86) and Child(90). We assume that it is harder to create a meaningful

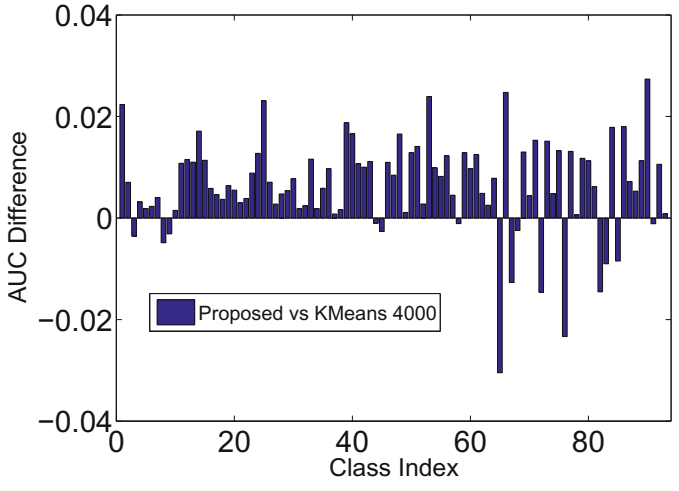


Fig. 5. Class-wise differences by AUC for CLEF2010, rgb channel between proposed method, 8000 words and vanilla k-means, 4000 words (other 2 omitted for readability)

binarization of all concepts into two classes when the number of concepts increases from 20 in VOC2009 to 93 in ImageCLEF2010. Here using a multi-class classification with several classes of approximately equal size could turn out to be beneficial.

5.2 Runtime Considerations

For 8000 visual words the classifier training required merely about 10 minutes on a 128 CPU cluster when using liblinear [18]. The linear SVMs are comparably fast despite the involved optimization problem, because they run on a small local set of features only. The k-means distance computation step is a simpler algorithm but gets executed globally on the set of all features (800000 for rgb SIFT, 2.4 Mio for grey SIFT). A single core k-means implementation required two hours for each iteration of k-means in order to generate 4000 visual words. For 8000 words the time would roughly be doubled. Typically multi-core implementations of k-means can be used to speed up this process as well as the unsupervised part in our proposed method, however they tend to end at parallelizations to 8 CPUs or require specialized hardware to use more cores. The proposed approach allows the distribution of the supervised classification part to independent CPUs. It has been already mentioned in Section 2 that our algorithm requires during bag of word computation time for N visual words $N/2 + 1$ inner product evaluations compared N such steps for vanilla k-means. This theoretical claim is consistent with the average feature computation times we observed per image for rgb-SIFT: KM4000 takes 58.73 s, ERCF16x512 takes 7.8 s, hierarchical KM4000x2 requires 64.09, a bag of words feature based on

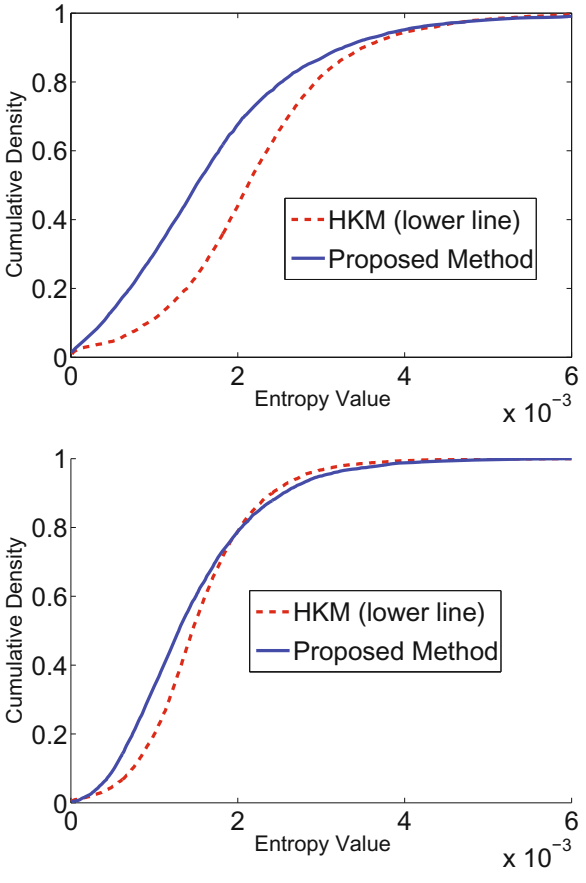


Fig. 6. Cumulative distribution functions of the label entropy across all visual words, rgb channel for the proposed method, 8000 words and hierarchical k-means, 8000 words, Upper: CLEF2010, Lower: VOC2009

KM8000 based on a prematurely terminated clustering needs 128.43 s. The running times for bag of words computation using such kind of clustering are fully comparable, whereas classification performance might be slightly degraded. The proposed method KM4000 + 4000 linear SVMs takes 68.28 s which is in the range of the other k-means based methods which include a vocabulary of size 4000.

In practice running times are affected by many additional factors. Our system for example stores sift features in bzip2-ed form for keeping disk space within reasonable bounds. This requires to uncompress them in memory which adds a certain offset to each bag of words computation step.

5.3 Label Distribution Statistics across Visual Words

We have seen in Section 5.1 that the proposed method performs better in terms of error measures. The performance was examined class-wise in the aforementioned section. Now we would like to assume the word-wise viewpoint. For sanity checking we examine whether the usage of local classifiers leads to a difference in base feature assignments to visual words. To this end, we computed for each visual word the entropy of label distribution given by all base features which are assigned to the visual word in question. Consider a visual vocabulary V and a set of K base features with $\{0,1\}$-valued multi-labels from C concepts $\{(b_i, y_{ji}), j = 1,\ldots,C, i = 1,\ldots,K\}$. Define using proper normalization

$$p(y_c, v) \propto \sum_{i|v=\arg\min_{w\in V} d(b_i,w)} y_{ci} \tag{4}$$

We can compute the corresponding entropy $\tau(v)$ of the label distribution for a fixed word [19].

$$\tau(v) = -\sum_{j=1}^{C} p(y_j|v)\log(p(y_j|v)) \tag{5}$$

A lower entropy suggests a better separation of base features assigned to different labels for a given word. To compare a visual vocabulary as a whole we consider the cumulative sums of distribution of these entropies over the set of all visual words in a vocabulary. Figure 6 compares this distribution between the proposed method and hierarchical k-means with 4000x2 clusters, which is structurally the closest baseline, both using 8000 words. We observe that the proposed method has a higher mass of visual words at lower values of the label distribution entropy, which indicates that more informative words about some visual concepts were selected.

6 Conclusion

In this paper, we proposed a hybrid algorithm of unsupervised clustering and supervised linear SVMs for visual codebook generation. On VOC2009 and ImageCLEF data sets, we showed clear advantages either in recognition performance or in computation speed over purely unsupervised (e.g. k-means, HKM) and purely supervised (e.g. ERCF) procedures. In particular, our approach can reduce runtime of word generation and assignment almost half without losing performance, while the fastest choice ERCF often loses unignorable amounts in performance scores. Furthermore, we can double the standard k-means vocabularies with small extra costs, which brought substantial improvements in VOC2009 and for majority of concepts in ImageCLEF2010. Further research should be done to incorporate fast linear multi-class classifiers or to find better binarization procedures used for the SVM training.

Acknowledgement. This work was supported in part by the German Federal Ministry of Economics and Technology (BMWi) under the project THESEUS (01MQ07018).

References

1. Csurka, G., Bray, C., Dance, C., Fan, L.: Visual categorization with bags of keypoints. In: Workshop on Statistical Learning in Computer Vision, ECCV, Prague, Czech Republic, pp. 1–22 (2004)
2. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge (VOC 2007) (2007), http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html
3. Tahir, M., van de Sande, K., Uijlings, J., Yan, F., Li, X., Mikolajczyk, K., Kittler, J., Gevers, T., Smeulders, A.: Surreyuva srkda method (2008), http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2008/workshop/tahir.pdf
4. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2009 (VOC 2009) (2009), http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html
5. Nowak, S., Dunker, P.: Overview of the CLEF 2009 large-scale visual concept detection and annotation task. In: Peters, C., Caputo, B., Gonzalo, J., Jones, G.J.F., Kalpathy-Cramer, J., Müller, H., Tsikrika, T. (eds.) CLEF 2009. LNCS, vol. 6242, pp. 94–109. Springer, Heidelberg (2010)
6. Jurie, F., Triggs, B.: Creating efficient codebooks for visual recognition. In: ICCV 2005, vol. I, pp. 604–610 (2005)
7. Moosmann, F., Triggs, B., Jurie, F.: Fast discriminative visual codebooks using randomized clustering forests. In: Advances in Neural Information Processing Systems (2006)
8. Moosmann, F., Nowak, E., Jurie, F.: Randomized clustering forests for image classification. IEEE Transactions on Pattern Analysis & Machine Intelligence 30, 1632–1646 (2008)
9. Uijlings, J., Smeulders, A., Scha, R.: Real-time bag-of-words, approximately. In: CIVR (2009)
10. Bosch, A., Zisserman, A., Muñoz, X.: Representing shape with a spatial pyramid kernel. In: Proceedings of the 6th ACM International Conference on Image and Video Retrieval (CIVR 2007), pp. 401–408 (2007)
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: Workshop on Generative-Model Based Vision (2004)
12. van de Sande, K.E.A., Gevers, T., Snoek, C.G.M.: Evaluating color descriptors for object and scene recognition. IEEE Trans. Pat. Anal. & Mach. Intel. (2010)
13. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: CVPR 2006: Proceedings of Conference on Computer Vision and Pattern Recognition (2006)
14. Fulkerson, B., Vedaldi, A., Soatto, S.: Localizing objects with smart dictionaries. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 179–192. Springer, Heidelberg (2008)
15. Lowe, D.: Distinctive image features from scale invariant keypoints. International Journal of Computer Vision 60, 91–110 (2004)
16. Sonnenburg, S., Rätsch, G., Henschel, S., Widmer, C., Behr, J., Zien, A., de Bona, F., Binder, A., Gehl, C., Franc, V.: The shogun machine learning toolbox. Journal of Machine Learning Research (2010)

17. van Gemert, J.C., Geusebroek, J.-M., Veenman, C.J., Smeulders, A.W.M.: Kernel codebooks for scene categorization. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 696–709. Springer, Heidelberg (2008)
18. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. J. Mach. Learn. Res. 9, 1871–1874 (2008)
19. Wojcikiewicz, W., Binder, A., Kawanabe, M.: Enhancing image classification with class-wise clustered vocabularies. In: Proceedings of the 20th International Conference on Pattern Recognition (ICPR) (2010)

Appendix: Results for Grey SIFT Features

Table 5. Results for VOC2009 dataset with grey SIFT, 10-fold cross-validation

Method / Score	AP	AUC
baseline: SIFT, hierarch KM250x2 words	43.27 ± 5.23	85.48 ± 1.93
baseline: SIFT, ERCF4x128 words	41.11 ± 5.21	84.84 ± 1.54
baseline: SIFT, KM500	44.03 ± 5.74	85.81 ± 1.90
proposed: SIFT, KM250, 5 iter+250 lin SVM	44.28 ± 5.75	**85.94** ± 1.91
proposed: SIFT, KM250 +250 lin SVM	**44.60** ± 5.18	85.93 ± 1.86
baseline: SIFT, hierarch KM4000x2	49.49 ± 5.03	87.71 ± 1.67
baseline: SIFT, ERCF16x512	52.04 ± 5.14	88.60 ± 1.71
baseline: SIFT, KM4000	51.28 ± 5.59	88.47 ± 1.47
proposed: SIFT, KM4000+4000 lin SVM	**52.07** ± 5.08	**88.85** ± 1.63

Table 6. Results for CLEF2010 PhotoAnnotation dataset with grey SIFT, 10-fold cross-validation

Method / Score	AP	AUC
baseline: SIFT, hierarch KM250x2	32.37 ± 1.04	73.49 ± 1.28
baseline: SIFT, ERCF4x128	31.48 ± 1.28	72.57 ± 1.84
baseline: SIFT, KM500 words	32.62 ± 0.96	73.57 ± 1.40
proposed: SIFT, KM250, 5 iters+250 lin SVM	32.31 ± 1.08	73.49 ± 1.41
proposed: SIFT, KM250, 120 iters+250 lin SVM	**32.78** ± 1.16	**73.82** ± 1.39
baseline: SIFT, hierarch KM4000x2	35.42 ± 1.09	75.78 ± 1.36
baseline: SIFT, ERCF16x512	34.82 ± 1.05	75.00 ± 1.50
baseline: SIFT, KM4000	35.13 ± 1.32	75.30 ± 1.32
proposed: SIFT, KM4000+4000 lin SVM	**35.74** ± 1.26	**76.04** ± 1.25

Image Inpainting Based on Probabilistic Structure Estimation

Takashi Shibata, Akihiko Iketani, and Shuji Senda

NEC Corporation, 1753 Shimonumabe, Nakahara-Ku, Kawasaki, Kanagawa 211-8666, Japan

Abstract. A novel inpainting method based on probabilistic structure estimation has been developed. The method consists of two steps. First, an initial image, which captures rough structure and colors in the missing region, is estimated. This image is generated by probabilistically interpolating the gradient inside the missing region, and then by flooding the colors on the boundary into the missing region using Markov Random Field. Second, by locally replacing the missing region with local patches similar to both the adjacent patches and the initial image, the inpainted image is synthesized. Since the patch replacement process is guided by the initial image, the inpainted image is guaranteed to preserve the underlying structure. This also enables patches to be replaced in a greedy manner, i.e. without optimization. Experiments show the proposed method outperforms previous methods in terms of both subjective image quality and computational speed.

1 Introduction

Inpainting is a technique for restoring damaged paintings and photographs by filling in missing regions with textures surrounding them. In computer vision, this technique has been applied for removing selected objects in images and has become one of the most active research areas in the field.

Various inpainting methods have been proposed. These methods can be classified into two categories: diffusion-based and exemplar-based methods. Diffusion-based methods [2,4] construct a diffusion equation for each pixel in the missing region that relates the color expected at the pixel and in its neighbor. Solving these equations, the colors surrounding the region are diffused into the region. Although these methods work effectively on relatively narrow regions, the result tends to be oversmoothed, especially in case of larger regions, since they assume the color smoothness inside the missing region.

On the other hand, exemplar-based methods [5,6,7,8,9,10,11,13,14] restore the missing region by pasting square patches sampled from the exterior of the region. Since the missing region is filled in the unit of a patch, which is large enough to capture textural patterns, the result is less likely to be oversmoothed. Exemplar-based methods can be further classified into two categories: greedy and globally optimal methods. Greedy methods iteratively paste patches into the missing region until no missing area is left. This method was first proposed by Harrison

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 109–120, 2011.


(a) original image


(b) greedy method [6]


(c) globally optimal method [14]

Fig. 1. Inpainted results by previous methods

et al. [7], and since then, various modifications have been proposed. Criminisi et al. introduced a "priority" measure that specifies which part of the missing region should be inpainted prior to the others [6]. Other methods improve the accuracy of patch selection [5,8,11,13]. Although these methods are faster than globally optimal methods (to be described later), they are inherently subject to a local minima problem due to their greedy procedure, often resulting in discontinuity in the inpainted region. An example is shown in Fig.1. In Fig.1(a), meshed regions show regions to be inpainted. The inpainted result synthesized by Criminisi et al. [6] is shown in Fig.1(b). Discontinuity is apparent on handrails and walls.

On the other hand, globally optimal methods treat inpainting as combinatorial optimization on patch selection [9,10,14]. These methods estimate the optimal combination of patches to fill in the missing region by minimizing the overall discontinuity within the region for a given set of patches. Patch selection is optimized by belief propagation [10] or by EM algorithm [14]. To avoid ambiguity in patch selection, Kawai et al. introduced an additional constraint that narrows down the search area for candidate patches [9]. These methods, in contrast to greedy methods, are capable of synthesizing continuous inpainted regions. This, however, does not ensure that the underlying structure in the missing region is preserved. An example of this, generated by Wexler et al. [14], is shown in Fig.1(c). Although a continuous inpainted region is obtained, the underlying structure (e.g. handrails, boundaries between the floor and the walls) is not preserved. Another issue with these methods is that the optimization process requires far more computation than greedy methods.

Some of the recently proposed methods focus on structure preserving inpainting. In Shift-Map editing [12], regions are copied into the missing region with uniform shifts. Since each region is uniformly shifted, structure within each region is expected to be preserved. However, the method often generates discontinuity on boundaries where different regions meet. PatchMatch [1] carries out structure-preserving inpainting by manually specifying the underlying structure. Although this method works well on a variety of images, the need for manually specifying the structure limits its application.

We propose an inpainting method that can preserve the structure underlying the missing region and is also computationally efficient. In the proposed method, first, an initial inpainted image, which captures rough structure and colors in the

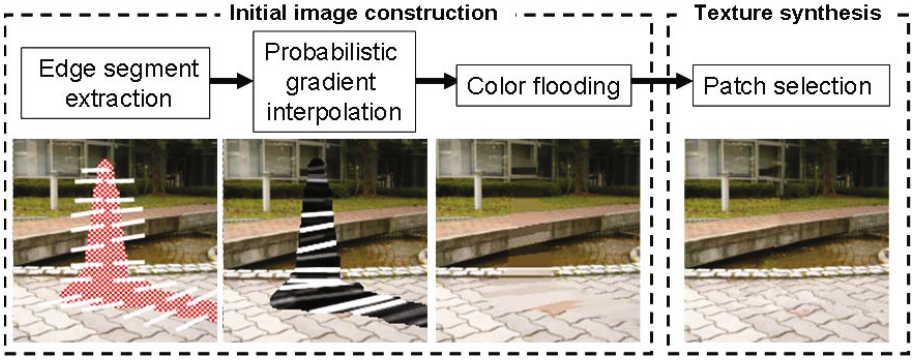


Fig. 2. Overview of proposed method

missing region, is estimated assuming the underlying structure consists of lines. Then, using this image as a guide, patches are sampled from the exterior of the region and pasted into the region iteratively. Patch selection guided by the initial image not only ensures the underlying structure is preserved but also enables the patches to be selected in a greedy manner, i.e. without optimization. This makes our method substantially faster than globally optimal methods [9,10,14], in which selected patches must be optimized.

2 Proposed Method

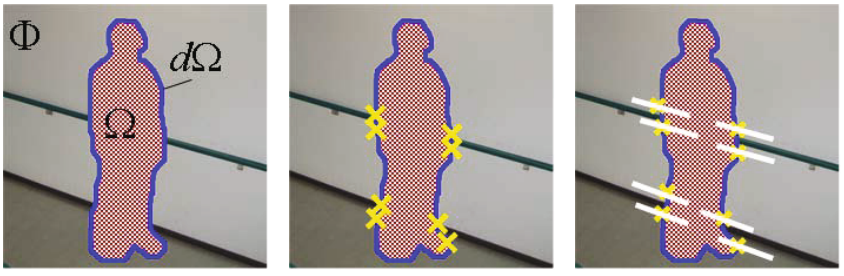
The overview of the proposed method is shown in Fig.2. As shown in Fig.2, the proposed method consists of two steps: initial image construction and texture synthesis.

In the first step, an initial image, which captures rough structure and colors in the missing region, is estimated. This image is generated by three processes: 1) extracting edge segments intersecting the boundary of the missing region, 2) probabilistically interpolating the gradient inside the region, and 3) flooding colors on the boundary of the region to the topographic relief formed by the gradient magnitude.

In the second step, patches are sampled from the exterior of the missing region and pasted into the missing region iteratively in a greedy manner. Each step is detailed below.

2.1 Initial Image Construction

Edge Segment Extraction. An example of an image with a missing region is shown in Fig.3(a). This step starts from extracting edge segments intersecting $d\Omega$, the boundary between the missing region Ω and the source region Φ. First, end points of the edge segments, depicted with $\times$ marks in Fig.3(b), are detected by searching local maxima of gradient along $d\Omega$. Then, Hough transform



(a) missing region (b) end points (c) edge segments

Fig. 3. Extracting end points and edge segments

is applied to Φ to detect edge segments intersecting the end points, as shown in Fig.3(c). Note that only the votes for the lines that intersect the end points are considered.

Probabilistic Gradient Interpolation. Next, the gradient inside Ω is interpolated. Here, the basic idea is to extend each edge segment into the missing region, and to assign the gradient of the edge segment to every pixel on the extended edge. Previous methods extend edge segments manually [1,13] or automatically based on some heuristics [5,8]. In general, however, it is unknown to what extent they should be extended. For example, in Fig.4(a), obviously edge segments on one side of the boundary should be fully extended to the other side, whereas in Fig.4(b), edge segments should be terminated where they intersect with other edges. These examples suggest that 1) if an end point has a corresponding point on the other side of the boundary, the edge segment is likely to be fully extended, and 2) the more intersections an edge encounters, the less likely it is to be further extended.

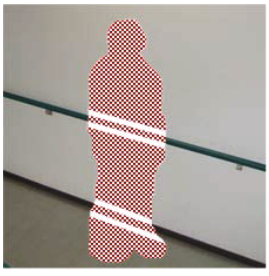
On the basis of this idea, we probabilistically determine to what extent each edge segment should be extended. We refer to the former type of edges as full edges and the latter as half edges. First, for each edge segment detected in the previous process, we consider two hypotheses: one for being a full edge and the other for being a half edge. Then, for every pixel in the missing region, the likelihood for belonging to each hypothetical edge is computed. Finally, each pixel is determined to belong to an edge that gives maximum likelihood. Further details are described below.

Let us begin by defining the two likelihoods. The former likelihood, i.e. the likelihood that a pixel at $\mathbf{x}$ belongs to a full edge connecting end points $\mathbf{x}_k$ and $\mathbf{x}_l$, is given by

$$L_{full}(\mathbf{x}, k, l) = L_0 \exp[-\frac{d_{kl}^2(\mathbf{x})}{2\sigma_d} - \frac{p(\mathbf{x}_k, \mathbf{x}_l)}{2\sigma_p}] \quad (1)$$

where $d_{kl}(\mathbf{x})$ is the distance from $\mathbf{x}$ to the edge connecting $\mathbf{x}_k$ and $\mathbf{x}_l$(see Fig.5), and $p(\mathbf{x}_k, \mathbf{x}_l)$ is the dissimilarity between end points at $\mathbf{x}_k$ and $\mathbf{x}_l$, which increases as the differences in gradient and edge direction become larger, i.e.,

(a) full edges



(b) half edges

Fig. 4. Two types of edges considered

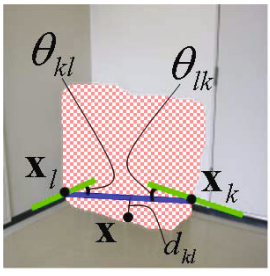


Fig. 5. Definition of θ_{kl},θ_{lk} and d_{kl}

$$p(\mathbf{x}_k, \mathbf{x}_l) = \sum_{i=R,G,B} \|\nabla I_i(\mathbf{x}_k) - \nabla I_i(\mathbf{x}_l)\|_2 + \lambda(\theta_{lk}^2 + \theta_{kl}^2) \tag{2}$$

where $\nabla I_i(\mathbf{x})$ is the gradient at $\mathbf{x}$ for each color component $i = R, G, B$, and θ_{lk} (or θ_{kl}) is the angle formed by two lines: the line connecting end points at $\mathbf{x}_k$ and $\mathbf{x}_l$, and the edge segment detected at $\mathbf{x}_k$ (or $\mathbf{x}_l$). σ_d, σ_p, L_0 and λ are parameters determined empirically.

The latter likelihood, i.e. the likelihood that $\mathbf{x}$ belongs to a half edge starting from an end point $\mathbf{x}_k$, is given by

$$L_{half}(\mathbf{x}, k) = w^{-n} \exp[-\frac{d_k^2(\mathbf{x})}{2\sigma_d}] \tag{3}$$

where w is a constant less than 1, n is the number of intersections with other edges encountered between $\mathbf{x}_k$ and $\mathbf{x}$, and $d_k(\mathbf{x})$ is the distance from $\mathbf{x}$ to the edge from $\mathbf{x}_k$. Note that each time the edge intersects another, this likelihood diminishes by a factor of w.

These likelihoods are used to determine to which edge each pixel $\mathbf{x}$ belongs. First, assuming that $\mathbf{x}$ belongs to a full edge, $L_{full}(\mathbf{x}, k, l)$ is computed for every pair of edge segments k, l, and the full edge that $\mathbf{x}$ most likely belongs to is determined by maximum likelihood estimation:

$$k_{\mathbf{x}}^{full}, l_{\mathbf{x}}^{full} = \arg\max_{k,l} L_{full}(\mathbf{x}, k, l) \tag{4}$$

where $k_{\mathbf{x}}^{full}$,$l_{\mathbf{x}}^{full}$ denote the two end points of the full edge.

Similarly, assuming that $\mathbf{x}$ belongs to a half edge, $L_{half}(\mathbf{x}, k)$ is computed for every edge segment k, and the half edge to which $\mathbf{x}$ most likely belongs is estimated as follows:

$$k_{\mathbf{x}}^{half} = \arg\max_{k} L_{half}(\mathbf{x}, k) \tag{5}$$

where $k_{\mathbf{x}}^{half}$ denotes the end point of the half edge.

Given both likelihoods, we finally determine whether $\mathbf{x}$ belongs to a full or a half edge, and estimate the gradient for $\mathbf{x}$. Here again, maximum likelihood estimation is used. The gradient $\nabla\tilde{I}(\mathbf{x})$ is given as the intensity gradient of the edge to which it belongs, weighted by the likelihood, i.e.

$$\nabla\tilde{I}(\mathbf{x}) = \begin{cases} \nabla I_{\mathbf{x}}^{full} L_{full}(\mathbf{x}, k_{\mathbf{x}}^{full}, l_{\mathbf{x}}^{full}) & if\ L_{full}(\mathbf{x}, k_{\mathbf{x}}^{full}, l_{\mathbf{x}}^{full}) \geq L_{half}(\mathbf{x}, k_{\mathbf{x}}^{half}) \\ \nabla I_{\mathbf{x}}^{half} L_{half}(\mathbf{x}, k_{\mathbf{x}}^{half}) & else \end{cases} \tag{6}$$

where $\nabla I_{\mathbf{x}}^{full}$ is the average of intensity gradient vectors at $\mathbf{x}_k$ and $\mathbf{x}_l$, and $\nabla I_{\mathbf{x}}^{half}$ is the gradient at $\mathbf{x}_k$.

Color Flooding. An initial image is generated by flooding colors on $d\Omega$ to the topographic relief formed by the gradient magnitude.

We formulate this problem on Markov Random Field (MRF). We consider a regular grid on the original image that covers the entire Ω and its neighbors in Φ. Each node corresponds to a pixel in the image. The energy function to be minimized is given as follows:

$$E(\mathbf{M}) = E_{data}(\mathbf{M}) + E_{smooth}(\mathbf{M}) \tag{7}$$

where $\mathbf{M} = \{M_1, M_2, \cdots, M_p, \cdots, M_N\}$ is a set of labels assigned to each of the N nodes in MRF, and M_p is a label given to the node at $\mathbf{x}_p$. Each label is uniquely associated with a color by function $f(M_p)$. Note that the color space is quantized to 64 levels in order to decrease the number of possible labels, thereby making the problem more tractable. The data term in the above function is defined as follows:

$$E_{data}(\mathbf{M}) = \sum_{p \in \Phi} \|f(M_p) - T(\mathbf{x}_p)\|_2 \tag{8}$$

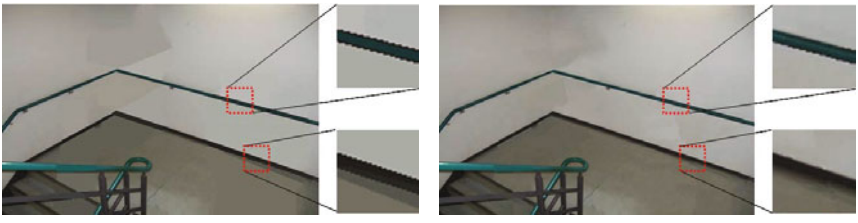
where $T(\mathbf{x}_p)$ is the color at $\mathbf{x}_p$ in the original image. This term penalizes the estimated colors deviating from its original colors. Note that this term is only computed at nodes in Φ, where original colors are available. The smoothness term is defined as follows:

$$E_{smooth}(\mathbf{M}) = \sum_{p,q \in R} V_{pq} \|f(M_p) - f(M_q)\|_2 \tag{9}$$

where R is a set of adjacent nodes. V_{pq} is a weight for the smoothness term, which is designed such that the closer the direction from $\mathbf{x}_p$ to $\mathbf{x}_q$ is to the gradients at $\mathbf{x}_p$ and $\mathbf{x}_q$, the lighter it becomes, i.e.,

$$V_{pq} = \exp[-\alpha|\mathbf{n}_{pq} \cdot (\nabla\tilde{I}(\mathbf{x}_p) + \nabla\tilde{I}(\mathbf{x}_q))|] \tag{10}$$

where $\mathbf{n}_{pq}$ is the unit vector the direction of which coincides with that from $\mathbf{x}_p$ to $\mathbf{x}_q$, $\cdot$ denotes the inner product of vectors, and α is a parameter determined empirically. This weight adds an edge-preserving property to the smoothness term, where abrupt color transition is tolerated along the gradient, i.e. across the edge, while imposing color uniformity in the other directions.



(a) Initial image (b) Final inpainted image

Fig. 6. Result synthesized by proposed method

The suboptimal solution for this problem is obtained by graph cuts [3]. The initial image generated by this process for the original image in Fig.1(a) is shown in Fig.6(a). While the image recovers the underlying structure in the missing region, close shots on the right seem somewhat posterized. This will be corrected by the subsequent patch selection process.

2.2 Patch Selection

In the second step, patches are sampled from Φ and pasted into Ω iteratively in a greedy manner. Here, a patch centered at $\mathbf{x}_j$ in Φ is selected as a patch to be pasted to $\mathbf{x}_i$ in accordance with the following criteria: 1) a patch at $\mathbf{x}_j$ is continuous to the already pasted patches in the neighbors of $\mathbf{x}_i$, and 2) is similar to the corresponding region in the initial image. Formally,

$$\mathbf{x}_j = \arg\min_{\mathbf{x}_q \in \Phi} c_{adj}(\mathbf{x}_i, \mathbf{x}_q) + c_{init}(\mathbf{x}_i, \mathbf{x}_q) \tag{11}$$

where $c_{adj}(\mathbf{x}_i, \mathbf{x}_q)$ is the sum of squared differences (SSD) between the patch at $\mathbf{x}_i$ in the inpainted image under construction and the patch at $\mathbf{x}_q$, and $c_{init}(\mathbf{x}_i, \mathbf{x}_q)$ is SSD between the patch at $\mathbf{x}_i$ in the initial image and the patch at $\mathbf{x}_q$. Note that in $c_{adj}(\mathbf{x}_i, \mathbf{x}_q)$, difference is computed only at pixels already filled by previously pasted patches.

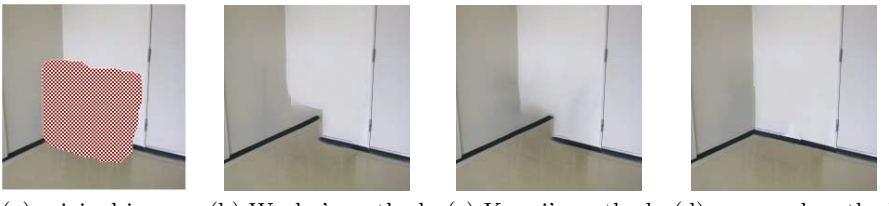
The final result obtained by this process is shown in Fig.6(b). As can be seen, smooth gradations induced by highlight and shade have been recovered.

Table 1. Parameters used in experiments

Size of patches	7×7[pixel]	L_0	9.00
w	0.50	λ	1.25×10^{-1}
σ_p	6.125×10^2	α	1.67
σ_d	0.50		

(a) original image (b) Wexler's method (c) Kawai's method (d) proposed method

Fig. 7. Experiment on house image



(a) original image (b) Wexler's method (c) Kawai's method (d) proposed method

Fig. 8. Experiment on indoor image

(a) original image (b) Wexler's method (c) Kawai's method (d) proposed method

Fig. 9. Experiment on tree image

3 Experiments

We experimented on 70 images in an image inpainting dataset [15] collected by Kawai et al. for evaluating their method [9]. This section first presents the inpainted results obtained by the proposed method and previous methods. Then, these results are compared in terms of subjective image quality. Finally, computational time of the proposed methods is presented. Note that all the images used in this experiment have the resolution of 200×200 pixels, and were processed on a PC with 3.2GHz CPU, 2.0 GB RAM. Refer to Table 1 for the parameters used in the proposed method.

3.1 Inpainted Results

Figure 7 to Fig.10 show 4 of the 70 experimental results. Each figure shows the original image, results by Wexler's method [14], Kawai's method [9], and the proposed method in (a), (b), (c) and (d), respectively.

(a) original image (b) Wexler's method (c) Kawai's method (d) proposed method

Fig. 10. Experiment on temple image

(a) house (b) indoor (c) tree (d) temple

Fig. 11. Initial images generated by the proposed method

In Fig.7(b) and (c), linear structures in the inpainted regions fade away, resulting in blurry images. The same problem also occurs in Fig.7(b) and (c). On the other hand, in Fig.7(d) and Fig.8(d), the proposed method gives sharper images. The reason for this is that the proposed method successfully recovers the structures underlying the missing regions in constructing the initial images, shown in Fig.11(a) and (b), and these images are utilized as a guide in generating Fig.7(d) and Fig.8(d).

In Fig.9(b) and (c), previous methods erroneously select patches from the background gray area. Similarly in Fig.10(c) and (d), patches from the background trees are selected. On the other hand, in Fig.9(d) and Fig.10(d), the proposed method, guided by the initial images shown in Fig.11(c) and (d), successfully selects correct patches. These results also show the effectiveness of the two-step algorithm in the proposed method.

3.2 Subjective Evaluation of Image Quality

For all 70 images, results obtained by each method were evaluated by 4 subjects in terms of image quality. Each subject rated every result on a scale ranging from 1 (very bad) to 5 (very good). For each method, Table 2 shows the average rating (i.e. ratings averaged over every image and subject), the number of images that had average ratings equal to or higher than 4 (good), and the number of images that acquired the highest ratings among the 3 methods. On all 3 measures, the proposed method outperformed the other methods.

In some images, however, the proposed method performed more poorly than the others. Such an example is shown in Fig.12. Rating for each result is shown in

(a) original image (b) Wexler's method (3.5) (c) Kawai's method (4.5) (d) proposed method (2.0)

Fig. 12. Experiment on road image

Table 2. Evaluation of subjective image quality

	Average rating	Number of images with rating ≥ 4	Number of highest rated images
Wexler's method [13]	3.41	21	18
Kawai's method [9]	3.64	30	28
Proposed method	3.76	40	36

parenthesis. As can be seen, the proposed method fails to recover the texture on the stone wall in Fig.12(d). This is a typical failure that occurs when patches are too small to capture the periodicity of texture. Using larger patches can improve the result, as shown in Fig.14. Here, instead of 7×7 pixel patches, 15×15 pixel patches were used. Note that the texture on the stone wall is successfully recovered. This result implies a method is needed to find the optimal patch size automatically. Developing such an algorithm remains our future work. Learning optimal values for parameters shown in Table. 1 from a set of images will be another interesting issue.

Finally, we present an image where none of the methods were successful. Fig 13 shows the results for an image where a barrel is partially occluded by a person. Note that every method failed in reconstructing the rim of the barrel, which led to equally low ratings. The reason for this failure can be summarized into two points: 1) in pasting patches of the rim, they need to be rotated in accordance with the normal of the rim, which none of the methods currently does, and 2) a guide is needed that forces patches to be aligned on an arc. Although the proposed method guides the patch pasting process by means of an initial image, it assumes the underlying structure to be composed of lines. For nonlinear structures, the method would fail in constructing the initial image, as shown in Fig.15. Extending the method to deal with more general types of structures is our future work.

3.3 Computational Time

The average computational time of the proposed method for all 70 images is shown in Table 3. According to Kawai [9], the time for Wexler's method [14] and

(a) original image (b) Wexler's method (2.8) (c) Kawai's method (2.8) (d) proposed method (3.0)

Fig. 13. Experiment on barrel image

Fig. 14. Experimental result with larger patch size

Fig. 15. Initial image for barrel image

Table 3. Computational time of the proposed method

Processing	Running time [sec]
Edge segment extraction	3.9
Probabilistic gradient interpolation	0.4
Color flooding	8.0
Patch selection	14.8
Overall	27.1

Kawai's method [9] are at the order of 10^2 to 10^3 sec. This shows the proposed method outperforms the previous methods by a factor of 10.

4 Conclusion

A novel inpainting method based on probabilistic structure estimation has been developed. Experiments show the method is more than 10 times faster than previous methods, while achieving better image quality. Currently, the method assumes the structure in the missing region is composed of linear edges. Extending the method to deal with more general objects, e.g. those with curves, remains our future work.

References

1. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: Patchmatch: A randomized correspondence algorithm for structural image editing. In: Proc. of ACM SIGGRAPH, vol. 29 (2009)
2. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: Proc. of ACM SIGGRAPH, pp. 417–424 (2000)
3. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. on PAMI 23, 1222–1239 (2001)
4. Chan, F.T., Shen, J.: Non-texture inpainting by curvature-driven diffusions. J. of VCIR 12, 436–449 (2001)
5. Chen, Y., Luan, Y., Li, H., Au, C.O.: Sketch-guided texture-based image inpainting. In: Proc. of ICIP, pp. 1997–2000 (2006)
6. Criminisi, A., Pérez, P., Toyama, K.: Region filling and object removal by exemplar-based image inpainting. IEEE Trans. on IP 13, 1200–1212 (2004)
7. Harrison, P.: A Non-hierarchical procedure for re-synthesis of complex texture. In: Proc. of WSCG, pp. 190–197 (2001)
8. Jia, J., Tang, K.C.: Image repairing: robust image synthesis by adaptive ND tensor voting. In: Proc. of CVPR, pp. 643–650 (2003)
9. Kawai, N., Sato, T., Yokoya, N.: Image inpainting considering brightness change and spatial locality of textures. In: Proc. of VISAPP, vol. 1, pp. 66–73 (2008)
10. Komodakis, N., Tziritas, G.: Image completion using efficient belief propagation via priority scheduling and dynamic pruning. IEEE Trans. on IP 16, 2649–2661 (2007)
11. Li, R.B., Qi, Y., Shen, K.X.: An image inpainting method. In: Conf. on CAD and Computer Graphics, pp. 531–536 (2005)
12. Pritch, Y., Kav-Venaki, E., Peleg, S.: Shift-map image editing. In: Proc. of ICCV, pp. 151–158 (2009)
13. Sun, J., Yuan, L., Jia, J., Shum, Y.H.: Image completion with structure propagation. In: Proc. of ACM SIGGRAPH, pp. 861–868 (2005)
14. Wexler, Y., Shechtman, E., Irani, M.: Space-time completion of video. IEEE Trans. of PAMI 29, 463–476 (2007)
15. http://yokoya.naist.jp/research2/inpainting/

Text Localization and Recognition in Complex Scenes Using Local Features

Qi Zheng[1], Kai Chen[1], Yi Zhou[1], Congcong Gu[1], and Haibing Guan[2]

[1] School of Information Security Engineering, Shanghai Jiao Tong University
[2] Department of Computer Science and Engineering, Shanghai Jiao Tong University

Abstract. We describe an approach using local features to resolve problems in text localization and recognition in complex scenes. Low image quality, complex background and variations of text make these problems challenging. Our approach includes the following stages: (1) Template images are generated automatically; (2) SIFT features are extracted and matched to template images; (3) Multiple single-character-areas are located using segmentation algorithm based upon multiple-size sliding subwindows; (4) An voting and geometric verification algorithm is used to identify final results. This framework thus is essentially simple by skipping many steps, such as normalization, binarization and OCR, which are required in previous methods. Moreover, this framework is robust as only SIFT feature is used. We evaluated our method using 200,000+ images in 3 scripts (Chinese, Japanese and Korean). We obtained average single-character success rate of 77.3% (highest 94.1%), average multiple-character success rate of 63.9% (highest 89.6%).

1 Introduction

Our goal is to read text from an image in complex scenes. There are many applications for such a technology, for example, recognizing sign from natural scenes, recognizing book/CD cover, license plate recognition, image and video search engine and web mining.

However, variations of text due to differences in size, style, orientation, and alignment, as well as low image quality, complex background and deformation in complex scenes make text localization and recognition a challenging task.

Previous methods [6–9] often consist of following stages, as shown in Figure 1(a), (1) Text localization and extraction; (2) Preprocessing; (3) OCR recognition. Of note, every stage consists of multiple steps that each has its own algorithm and usually operates sequentially.

Local features [1–5], which are distinctive and robust to noise, complicated background, and many kinds of geometric and photometric deformations, have been applied successfully in a wide range of systems and applications, such as wide baseline matching, object recognition, image retrieval, building panoramas, and video data mining. Moreover, as Figure 2 shows, local features matching can be potentially extended to text recognition problems.

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 121–132, 2011.

Inspired by the success of utilizing local features in image matching, we describe a local-feature-based approach for text localization and recognition. Considering the difference between text recognition and general image matching, we improve several steps in our approach accordingly, (1) we develop a new template-build method to automatically generate template images while eliminate the influence of complex scenes; (2) we develop a voting algorithm and a geometric verification algorithm for optimizing matching results and locating text; (3) we develop a segmentation algorithm based upon multiple-size sliding sub-windows to handle multiple characters efficiently.

Our framework, as shown in Figure 1(b), is essentially simple by skipping many usual steps, such as normalization, binarization, layout analysis and OCR, which are required in OCR-based methods. Moreover, this framework is robust and applicable in complex scenes as only SIFT feature is used during the process.

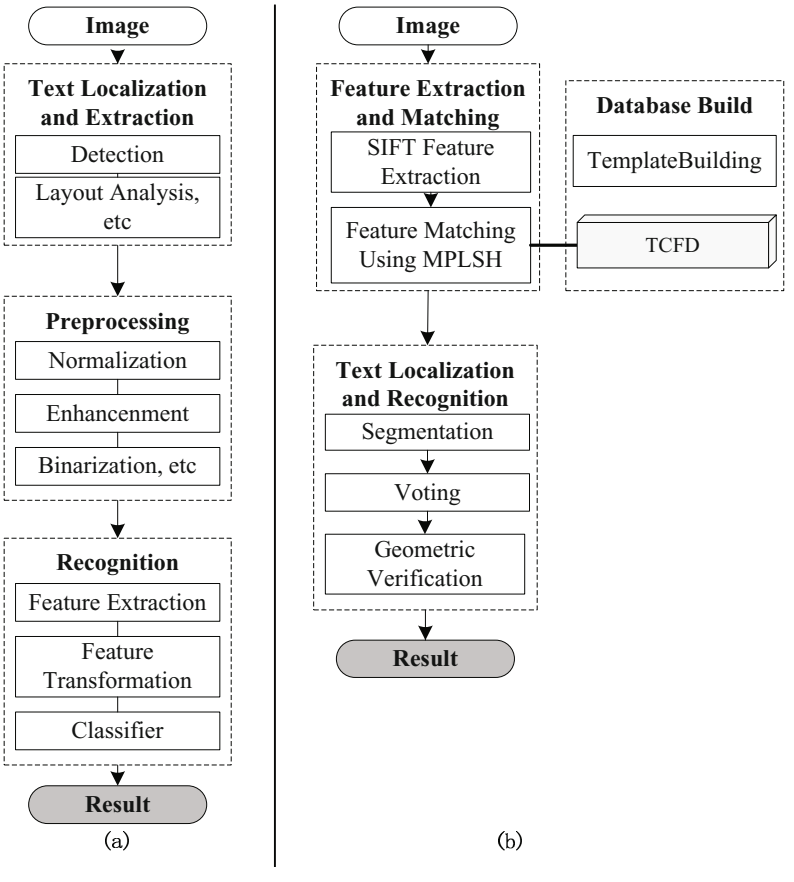


Fig. 1. Block diagram. (a) OCR-based framework; (b) Local feature-based framework. TCFD is template characters features database.

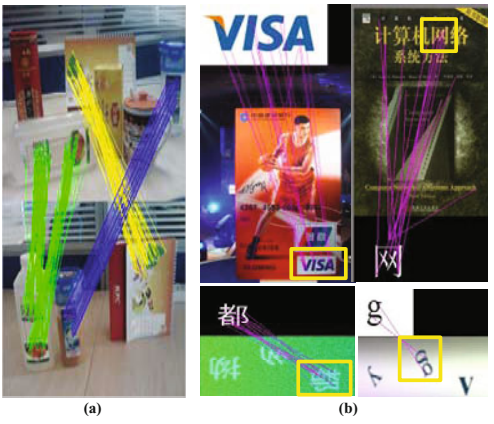


Fig. 2. Local features matching. (a) Object matching; (b) Characters matching.

1.1 Related Work

There have been a number of successful text localization and recognition works reported in [6–11]. Most of them follow the OCR-based framework. Chen et al [7] reported an approach of detection and recognition of sign from natural scenes. Laplacian of Gaussian (LOG) edge detector, color modeling, layout analysis and affine rectification are used to detect text. Then normalization is used as preprocessing. At last, intensity-based OCR is applied to recognize the text. Koga et al [9] introduced a camera-based Kanji recognition system for mobile-phones. The first stage consists of 4 steps: preliminary binarization, coarse layout analysis, line direction detection and line segmentation. The second stage consists of another 4 steps to identify the text: fine binarization, pre-segmentation, character classification and post processing. More detailed surveys can be found in [10–12].

Our approach is most similar to the work of Campos [13], which utilizes local features and bag-of-visual-words model (BoW) to recognize single character in English and Kannada. Yet the main differences between these two approaches are quite clear: (1) Our approach could handle the detection and recognition of multiple characters other than single character; (2) Template images are machine-generated instead of manually collected in our approach, providing tremendous convenience for Chinese and Japanese text recognition.

2 Local Feature-Based Approach

Our framework (Figure 1(b)) consists of four stages: (1) Template images are obtained automatically via our template-build method, then template characters SIFT feature database (TCFD) is built. (2) The SIFT features of query image are extracted and matched to TCFD using MPLSH. (3) Multiple single-character-areas are located based on our segmentation algorithm. (4) For each

single-character-area, a voting algorithm is used to identify candidate characters, which are then subjected to a geometric verification algorithm for final results. We describe these stages and methods in detail in the rest of this section.

2.1 Method of Building TCFD

Generation of template images for text matching is often challenged by the variation of characters (e.g. font, size, style). In some cases, the huge amounts of characters make the task even harder. For example, a total of 27474 characters are used in Chinese language compared with 26 letters in English.

In the field of image retrieval and object recognition, natural scene images are often used as template images. However, the local feature points in single character image are far less than that in a scene image. As a result, these interferences will greatly affect the matching accuracy if natural images are used as template images. Of note, for languages such as Chinese and Japanese, to obtain natural scene images will be indeed expensive and time consuming.

We applied the following strategies to build TCFD:

(1) The template images are machine-generated in monochrome mode without any additional noise and texture.
(2) According to fonts' similarity, a selected subset of fonts is used to generate template images per character.
(3) Every font per character will have two template images in TCFD (white-foreground/black-background or black-foreground/white-background) as shown in Figure 3(a). Using only one template image per font in some cases will result in zero matching points as shown in Figure 3(b). Furthermore, experimental results showed that using two template images readily gained 33.0% improvement over using one template image. Increasing the number of template images, however won't necessarily achieves further obvious improvement.

Table 1. Flowchart of Voting Algorithm

(1) Given an image, the initial vote of a candidate character is A, A = the number of matched features in query image. $A = 3$ in Figure 4(a).
(2) Given a candidate character, B is the number of matched features in the candidate image. $B = 6$ Figure 4(a).
(3) A is not always equal to B due to the similar matching. Then, the final vote is $V = Min(A, B)$. $V = 3$ in Figure 4(a).
(4) The template characters with top C votes are identified as final candidate images, $C = R/V$, we have chosen to use $R = 60$ and C is limit from 2 to 20. The more V is identified, the fewer candidates are retrieved.

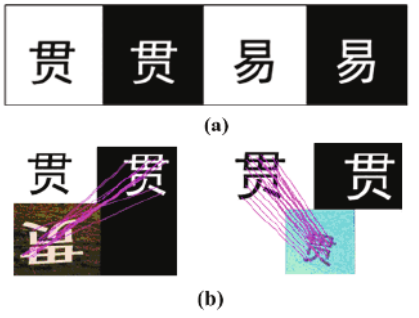


Fig. 3. Example of template images. (a) Two kinds of template images; (b) Only one template image has feature points matched for query images, so matching could be failed if only one template image is used.

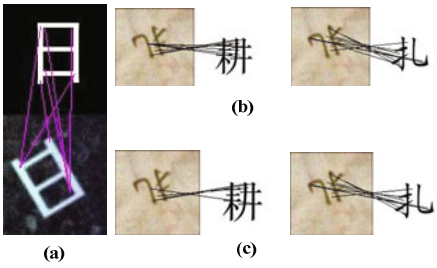


Fig. 4. Example images for voting algorithm and geometric verification algorithm. (a) Result before voting; (b) Result after voting; (c) Result after geometric verification.

2.2 Voting and Geometric Verification Algorithm

During text matching using local features, many mismatches can be caused by many factors, such as, similarities among characters, deformations and noises. Examples of mismatches could be found in Figure 4(a).

We designed a voting algorithm and gained 13.5% improvement. Optimized candidate characters are retrieved by using this algorithm. Flowchart of this algorithm is shown in Tab. 1.

Although the voting algorithm is helpful, there are still many mismatches in matching since local features are lack of global information. Such example results can be found in the left side of Figure 4(b).

Geometric verification can be used in character recognition for optimizing final results. This task however is often challenged by high computational cost and limited number of matched pairs.

Based upon the idea of pairwise constraint [16], we designed a geometric verification algorithm: Maximal Clique Matching for Text Recognition (MCM-TR). In MCM-TR, the global geometric constraint problem is expressed as the maximal clique problem in graph theory. MCM-TR starts from building a geometric correspondence graph (GCG) based upon the weak geometric constraint (WGC) information in local features. Then the global geometric relationship can be found by finding the maximal cliques in GCG. Given the characteristics of the global optimality of maximal cliques, MCM-TR is robust to occlusion, clutter, non-rigid deformations with the need of very few matched pairs.

We implemented MCM-TR as shown in Tab. 2 and achieved 9.5% improvement. The average matching time is 0.008 sec for two images with 60 matched pairs. Example results after geometric verification can be found in Figure 4(c).

Table 2. Flowchart of Geometric Verification Algorithm

(1) Given an image, and a candidate image identified by voting algorithm, all correspondences between these two images are labeled.
(2) For every matched pair, the space, scale and rotation information of SIFT features are extracted to estimate the WGC. To reduce the computational complexity, those match pairs whose points are not close in space and scale will be directly discarded.
(3) WGCs of all matched pairs are used to build GCG. GCG is an undirected and unweighted graph, in which each vertex represents a correspondence. The vertices are adjacent only when the correspondences are consistent with WGC. We make the projection from correspondences to GCG.
(4) The approximation algorithm proposed in [16] is extended to finding the maximal cliques in GCG. The maximal cliques just represent the global geometric relationship between the query image and the candidate image. To reduce the computational complexity, the maximal cliques containing too many or too few vertices are rejected.
(5) The candidate with max number of the matched pairs in the maximal clique is indentified as the final result.

2.3 Segmentation Algorithm and Multiple-Character Recognition

Segmentation algorithm is used to locate multiple single-character-areas in whole image. We call each area a sub-window, as Viola [17] use in face detection. We don't need to select all the feature points in the sub-window. The feature points of the same character are always similar in scale. We can filter the feature points by scale, which can greatly reduce the number of the local features. Furthermore, Hash table is used to rapidly obtain the local features in a sub-window.

Detail of algorithm is shown as following: Obtain the range of the location and scale of the local features matched in the MPLSH matching process. Let $W_{min} = S_{min}k$, $W_{max} = S_{max}k$. W_{min} and W_{max} represent the minimal and maximal size of the sub-window. The size of sub-windows increases by a factor of Δs between W_{min} and W_{max}. For each size, the sub-windows are shifted by some number of pixels $w\Delta l$. w is the size of sub-window. In each sub-window with size w, only those feature points whose scale is in the range of $(w/k, w\Delta s/k)$ are kept. The choice of Δl and Δd affects both the speed of recognition as well as accuracy. In this paper, $\Delta l = 2$ and $\Delta d = 0.5$.

Many sub-windows will be extracted in a query image. For example, in a 640x480 image, if the $W_{min} = 48$ and $W_{max} = 256$, 561 sub-windows will be extracted. It is high cost to recognize every sub-window.

The sub-windows will be subjected to the voting and geometric verification algorithm. However, it is not needed to recognize every sub-window. The reasons are: 1) there are few local features in some sub-windows. It is of low probability that there exist characters. 2) Some sub-windows cover the same region. If the characters in this region are recognized in one sub-window, the features of those characters do not need to be recognized anymore.

Table 3. Flowchart of Multiple Character Recognition Process

1. Statistic the number of the matched points in each sub-window. Those whose point number is less than threshold t will be removed from the heap. The sub-windows left are used to build a max heap.
2. While the point number of the top sub-window of the max heap is more than t:
2.1 Recognize the top sub-window. Let define the recognized character is C and the point number is n.
2.2 If $n < t$ and n is less than half number of template character, we determine there is no character in the sub-window. The sub-window will be removed from the heap.
2.3 If C is recognized, the accurate region and orientation of C can be computed by the transformation between the character and the template. The points of C are removed from the query image. Then update the point number of the sub-windows that cover C.
2.4 Update the max heap.
3. Combine overlapped recognized characters. Only the character with the most feature points will be left.

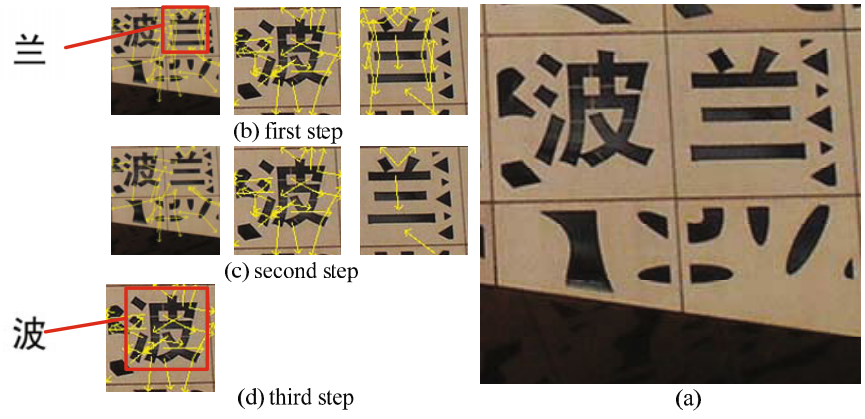


Fig. 5. An example of multiple-character recognition. (a) is the query image containing two characters; Figures in (b), (c) and (d) are the sub-windows extracted from the query image. The yellow lines represent local features. The left sub-window is selected for recognition. In the first step, a character is successfully recognized. The local features of the character will be removed in every sub-window. In the second step, no character is recognized. The left sub-window will be removed. The right sub-window will be removed because of too few points. In the third step, the other character is correctly recognized. The selected sub-window will be removed because of too few points left.

The multiple characters recognition process is shown in Tab. 3. An example is shown in Figure 5.

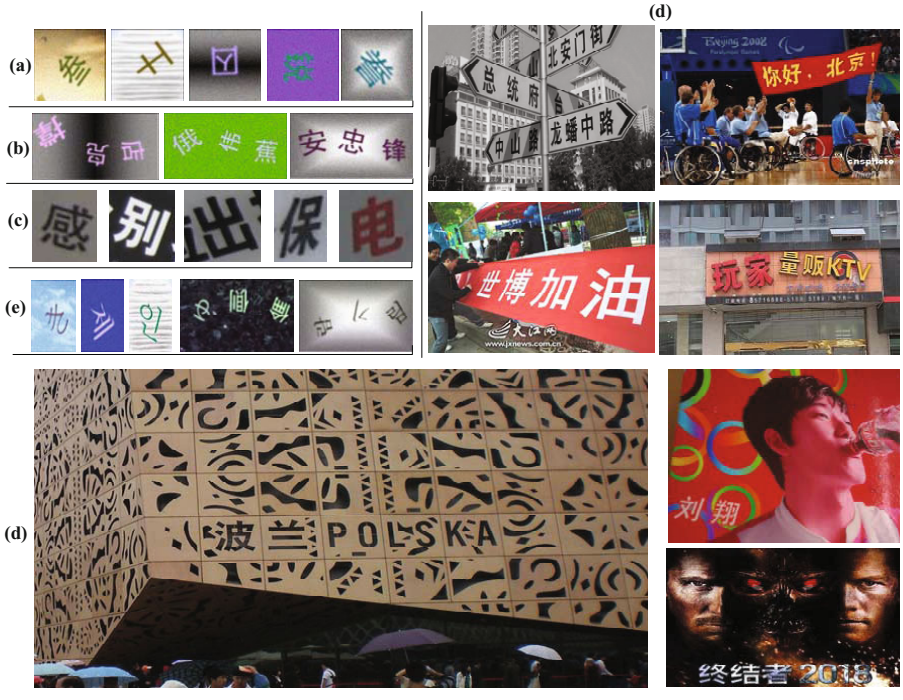


Fig. 6. Datasets. Examples of images used for the evaluation. (a) Dataset-A: Single Chinese characters; (b) Dataset-B: Multiple Chinese characters; (c) Dataset-C: Single Chinese characters from images of natural scenes; (d) Dataset-D: Multiple Chinese character images from natural scenes; (e) Dataset-E: Single and multiple characters of 3 languages scripts.

3 The Datasets

We built 5 datasets for the test: dataset-A, dataset-B, dataset-C and dataset-D are datasets containing only Chinese characters, dataset-E contains 3 language scripts (Chinese, Japanese and Korean). A summary of these five datasets is listed in Tab. 4. Examples of each dataset are shown in Figure 6.

4 Experimental Results

We performed 5 tests to evaluate our approach. Descriptions of these tests are shown in Tab. 4. In our experiments, we use Andrea Vedaldi's sift++[1] and Wei Dong's LSHKIT[2] for our SIFT and MPLSH [14] implementation.

The results of Test-1, Test-2, and Test-3 are shown in Tab. 5 and Tab. 7 accordingly. For Chinese text in complex scenes, we obtained average success

[1] http://www.vlfeat.org/~vedaldi/code/siftpp.html
[2] http://lshkit.sourceforge.net/index.html

Table 4. Description of each test

Tests	Datasets	Dataset Description	Objective of Test
Test-1	Dataset-A	3500 Single Chinese Characters; 168,000 machine-generated testing images in 12 fonts; 3 fonts in template characters.	To evaluate the success rate of single Chinese character, and describe the effect of various algorithms.
Test-2	Dataset-B	Multiple Chinese characters; 36,000 machine-generated test images in 12 fonts; 3 fonts in template characters.	To evaluate the success rate of multiple Chinese characters.
Test-3	Dataset-C	Single Chinese character image; Obtained from natural images; 1,000 testing images; 3 fonts in template characters.	To evaluate the success rate of single Chinese character from natural scenes, and compare to commercial OCR.
Test-4	Dataset-D	Multiple Chinese characters obtained from natural scenes; 120 Hei-like-font test images; 2 fonts in template characters.	To evaluate the success rate and false rate of multiple Chinese characters from natural scene images.
Test-5	Dataset-E	Chinese, Japanese, Korean; Single character and multiple characters; Machine generated; 15,700 testing images in 1 font; 1 font in template characters.	To evaluate the success rate of single character and multiple characters in 3 languages scripts.

rate of 77.3% (highest 94.1%) for single character and average success rate of 63.9% (highest 89.6%) for multiple characters. Compared with commercial OCR software, our approach improved the recognition accuracy by 12.9%. Given the character images all vary in fonts, lighting conditions, rotation, scale and affine deformation, these results are indeed encouraging.

There is 33.0% improvement by using our method to build the template images, 9.5% improvement by the geometric verification algorithm. The results demonstrated the efficiency of each steps of our approach.

It is also quite obvious from our studies that depending on the used fonts, the accuracy of text recognition changes dramatically too (e.g. the lowest rate 56.7% in the case of FangSong font). It indicated the importance of the selection of fonts in template images.

The results of Test-5 are shown in Tab. 6. Our approach also achieves encouraging results for language scripts other than Chinese script. We found the more SIFT points in a character images (the more complicated structure), the higher success rate.

The results of Test-4 are shown in Tab. 8. Results show that our approach is robustness in complex scenes. Some positive results are in Figure 7(a). We also found the success rate decreased for multiple characters from both natural scenes and machine-generated images. The main reasons are:

Table 5. Results of Test-1 and Test-2 (success rate of single Chinese character and multiple Chinese characters). E-1 is the approach without template-build method, E-2 is the approach without geometric verification algorithm. E-Single represents single character. E-Multi represents multiple characters.

Fonts	E-1	E-2	E-Single	E-Multi
Hei	53.3%	90.6%	94.1%	89.6%
MSYaHei	45.1%	70.4%	78.5%	63.2%
XiHei	49.4%	74.9%	84.3%	74.3%
PingHei	44.0%	65.4%	75.7%	66.7%
DengXian	48.8%	76.3%	84.6%	62.8%
YouYuan	37.1%	43.0%	58.7%	56.2%
GWArial	47.9%	80.6%	87.3%	66.1%
Song	44.5%	66.6%	76.4%	60.3%
FangSong	30.8%	40.8%	56.7%	50.7%
Kai	48.3%	81.2%	87.2%	67.8%
STKai	42.8%	67.0%	76.4%	60.2%
BWKai	39.7%	56.4%	67.5%	48.6%
Average	44.3%	67.8%	77.3%	63.9%

Table 6. Test results of Test-5

Language	Chinese	Japanese	Korean
Average number of SIFT points	60.3%	40.4%	28.5%
Success rate of single character	94.1%	88.3%	78.5%
Success rate of multiple characters	89.6%	77.1%	70.5%

Table 7. Test results of Test-3

Methods	Our Approach	Hanwang-Wenhao OCR 7600	Tsinghua-OCR 9.0 Pro
Success Rate	73.1%	60.2%	44.7%

Table 8. Test results of Test-4. **SROC:** the rate of the correct recognized characters among all the characters. **FROC:** the rate of false characters among all the recognized ones. **SROI:** The rate of the images with all characters correctly recognized and no false ones. **FROI:** the rate of those images with no characters correctly recognized.

Language	SROC	FROC	SROI	FROI
Chinese	60.4%	30.6%	12.5%	15.8%

(1) The complex change in natural scenes, such as joined characters (Figure 7(b)), varied foreground (Figure 7(c)), shadows, vertical characters, outline characters, large deformation (Figure 7(d)) and large illumination change will lead

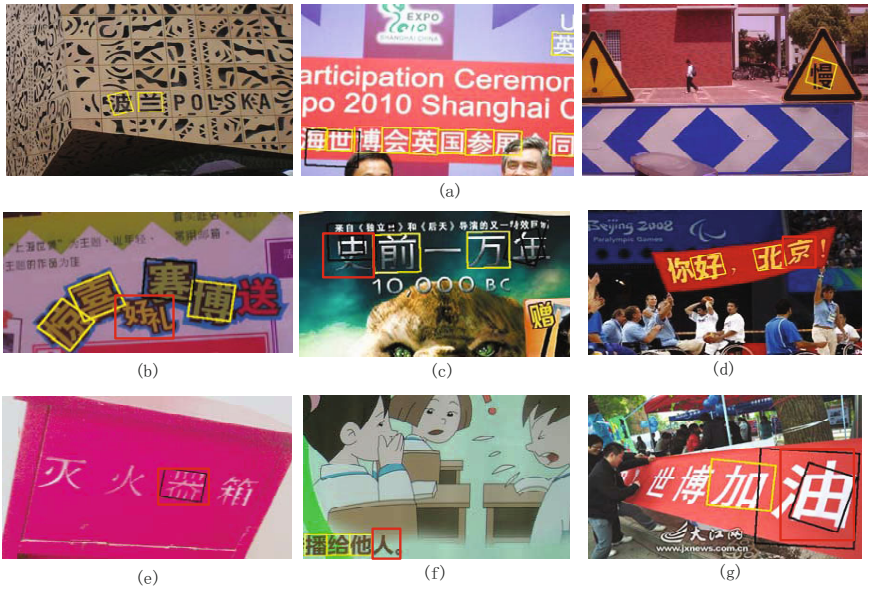


Fig. 7. Datasets. Examples of images used for the evaluation. (a) Dataset-A: Single Chinese characters; (b) Dataset-B: Multiple Chinese characters; (c) Dataset-C: Single Chinese characters from images of natural scenes; (d) Dataset-D: Multiple Chinese character images from natural scenes; (e) Dataset-E: Single and multiple characters of 3 languages scripts. Yellow rectangles represent correct recognition. Black ones represent false recognition. Red rectangles are manually drawn for further explanation.

to the great change in the scale, orientation and description of the feature points, which result in rejection of many matching pairs in both matching and geometric verification process. Moreover, low resolution (Figure 7(e)) and too thin strokes will cause very few SIFT feature detected.

(2) Threshold t will rejected characters with simple structures as well as the background noises. It is tradeoff between success rate and false rate. Figure 7(f) is the sample image that a simple character is rejected.

(3) Similar characters possibly received more votes than the character itself even after geometric verification, if wrong sub-windows are extracted and selected. In Figure 7(g), the selected sub-windows are either too big or too small.

5 Conclusion

In this paper, we describe a local-feature-based framework for text localization and recognition by only using SIFT features. The essential components in this framework include template-character-feature database buildup, a segmentation algorithm, a voting algorithm and a geometric verification algorithm. Our results demonstrated this approach performed well for texts in complex scenes, especially for those language scripts with complicated structures.

Although the robust performance of our approach suggests the local features matching can be utilized to address common problems in text localization and recognition, more works are needed toward a mature application. We plan to investigate other geometric verification methods and local features for better recognizing multiple characters with simple structure. We will explore such an approach in our future work.

Acknowledgement. This work was supported by National Natural Science Foundation of Shanghai (Grant No.10ZR1416400).

References

1. Lowe, D.: Distinctive image features from scale-invariant keypoints. In: ICCV, vol. 2, pp. 91–110 (2004)
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. Ke, Y., Sukthankar, R.: PCA-SIFT: A more distinctive representation for local image descriptors. In: CVPR, vol. 2, pp. 506–513 (2004)
4. Mikolajczyk, K., Schmid, C.: A Performance Evaluation of Local Descriptors. In: CVPR, vol. 2, pp. 257–263 (2003)
5. Tuytelaars, T., Mikolajczyk, K.: A Survey on Local Invariant Features. In: Foundations and Trends in Computer Graphics and Vision (2008)
6. Chen, X., Yuille, A.: Detecting and Reading Text in Natural Scenes. In: CVPR, vol. 2, pp. 366–373 (2004)
7. Chen, X., Yang, J., Zhang, J., Waibel, A.: Automatic detection and recognition of signs from natural scenes. IEEE Transactions on Image Processing 13, 87–99 (2004)
8. Chang, S.L., Chen, L.S., Chung, Y.C., Chen, S.W.: Automatic License Plate Recognition. IEEE Transactions on Intelligent Transportation Systems 5, 42–53 (2004)
9. Koga, M., Mine, R., Kameyama, T., Takahashi, T., Yamazaki, M., Yamaguchi, T.: Camera-based Kanji OCR for mobile-phones: practical issues. In: ICDAR (2005)
10. Liang, J., Doermann, D., Li, H.: Camera-based analysis of text and documents: A survey. IJDAR 7, 84–104 (2005)
11. Jung, K., Kim, K.I., Jain, A.K.: Text information extraction in images and video: a survey. Pattern Recognition 37, 977–997 (2004)
12. Fujisawa, H.: Forty years of research in character and document recognition - an industrial perspective. Pattern Recognition 41, 2435–2446 (2008)
13. de Campos, T.E., Babu, B.R., Varma, M.: Character recognition in natural images. In: VISAPP (2009)
14. Lv, Q., Josephson, W., Wang, Z., Charikar, M., Li, K.: Multi-probe LSH: Efficient indexing for high-dimensional similarity search. In: VLDB, pp. 950–961 (2007)
15. Johnson, D.S.: Approximation algorithms for combinational problems. JCSS 9, 256–278 (1974)
16. Leordeanu, M., Hebert, M.: A Spectral Technique for Correspondence Problems Using Pairwise Constraints. In: ICCV, vol. 2, pp. 1482–1489 (2005)
17. Viola, P., Jones, M.: Rapid Object Detection using a Boosted Cascade of Simple Features. In: CVPR, pp. 511–218 (2001)

Pyramid-Based Multi-structure Local Binary Pattern for Texture Classification

Yonggang He, Nong Sang, and Changxin Gao

Institute for Pattern Recognition and Artificial Intelligence
Huazhong University of Science and Technology
Wuhan, Hubei 430074, China

Abstract. Recently, the local binary pattern (LBP) has been widely used in texture classification. The conventional LBP methods only describe micro structures of texture images, such as edges, corners, spots and so on, although many of them show a good performance on texture classification. This situation still could not be changed, even though the multiresolution analysis technique is used in methods of local binary pattern. In this paper, we investigate the drawback of conventional LBP operators in describing some textures that has the same small structures but differential large structures. And a multi-structure local binary pattern operator is achieved by executing the LBP method on different layers of image pyramid. The proposed method is simple yet efficient to extract not only the micro structures but also the macro structures of texture images. We demonstrate the performance of our method on the task of rotation invariant texture classification. The experimental results on Outex database show advantages of the proposed method.

1 Introduction

Texture classification has been extensively investigated during the last several decades. Some methods for texture classification focus on the statistical analysis of texture images. The representative methods include the co-occurrence matrix method [2] and filtering based approaches [5, 11, 15]. Varma and Zisserman [16] present a good statistical algorithm, MR8, which uses 38 filters to build a rotation invariant texton library from a training set for classifying an unknown texture image. Recently, a simple but more powerful operator the local binary pattern (LBP) [13] that is based on the signs of differences of neighboring pixels is used for image description. And it has been successfully applied to texture analysis [10]. For texture classification, Ojala et al. [12] show a good performance of LBP for texture classification by comparing with other methods. And Mäenpää et al. [8] introduce a uniform pattern to robust texture description by selecting subsets of patterns encoded in LBP forms. With this technique, they propose a rotation invariant uniform pattern (LBP^{riu2}) [14] to describe the texture image. By utilising the temporal domain information, Zhao and Pietikäinen [17] extend the LBP to the VLBP for dynamic texture classification. Ahonen et al. [1] use the local binary pattern histogram Fourier features (LBP-HF) to describe rotation texture. Guo et al. [3] take the local variance as a weight of local binary

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 133–144, 2011.

pattern to adjust the contribution of the LBP code in histogram calculation and propose the LBPV operator for rotation invariant texture classification. Liao et al. [6] use the 80% dominant local binary pattern (DLBP) to classify the texture. And combining with Gabor features, it attains a high classification rate. The LBP-HF, LBPV and DLBP are both state-of-the-art algorithms and yield good results in the task of rotation invariant texture classification.

Although these operators perform well, most of them base on the same idea of LBP which only extracts isotropic micro structures of images. These micro structures are not enough to describe the texture information. The problem still can't be solved by the multiresolution LBP method [14] that just combines the limited neighbor sample points and radii. At the same time, the stability of LBP value deteriorates rapidly with the increasing of neighbor radius, because the sampling points have less correlation with the centre pixel with the present of larger radius. Multi-scale binary patterns (LBPF) [9] employs exponentially growing circular neighborhoods with Gaussian low-pass filtering to extract binary patterns for texture analysis. The LBPF also shows isotropic micro structures of images but a little bigger than the structures extracted by basic LBP methods. Our work considered the structures extracted by basic LBP. We carried out the rotation invariant uniform pattern LBP in an image pyramid to extract both micro and macro structures of texture images. The pyramid technique had been used in texture field by Heeger and Bergen [4] many years ago, but they focused on texture synthesis. In our work, four anisotropic filter templates ensured the collection of anisotropic structures in the image pyramid. Later, weights of different structural histograms were set to enhance the performance of proposed method. The results of experiment on Outex database show the superiority of our method.

The rest of this paper is organized as follows. Section 2 gives a brief overview of the basic LBP method and discusses the structures extracted by the conventional LBP. Section 3 devotes to the details of the proposed method. Section 4 presents the implementation of our experiments. Section 5 concludes his paper.

2 Local Binary Pattern

In this section, we review the LBP methods and points out the structures of texture images that are neglected by the conventional LBP. This is necessary for understanding the advantage of our method.

2.1 The LBP Methods

The local binary pattern (LBP) [14] is an illumination invariant texture operator which characterizes the local structure of the texture image. The basic LBP considers a small circularly symmetric neighborhood that has P equally spaced pixels on a circle of radius R. The LBP value of the center pixel is computed by thresholding the gray value of P sampling point with their center value, and summing the thresholded values weighted by powers of two. Thus, the LBP label for the center pixel (x,y) is obtained by

$$LBP_{P,R}(x,y) = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \tag{1}$$

$$s(x) = \begin{cases} 1, & x \geq 1 \\ 0, & x < 0 \end{cases} \tag{2}$$

where g_c is the gray value of the center pixel, g_p (p=0,...,*P-1*) correspond to the gray values of P sampling points. If the coordinates of g_c are (0,0), then the coordinates of g_p are given by $(-Rsin(2\pi p/P), Rcos(2\pi p/P))$. The gray values of neighbors which do not fall exactly in the center of grids are estimated by interpolation.

The rotation invariant version of LBP is achieved with the uniform measure. Mäenpää et al. [8] first defined the nonuniformity measure U ('pattern') as the number of transitions (0/1 or 1/0 changes) in the circular bitwise presentation of the LBP code. Later, Ojala et al. [14] designated patterns that have U value of at most 2 as 'uniform' and propose a rotation invariant uniform pattern operator $LBP^{riu2}_{P,R}$ for texture description:

$$LBP^{riu2}_{P,R} = \begin{cases} \sum_{p=0}^{P-1} s(g_p - g_c), & \text{if } U(LBP_{P,R}) \leq 2 \\ P+1, & otherwise \end{cases} \tag{3}$$

where

$$U(LBP_{P,R}) = \left|s(g_{P-1} - g_c) - s(g_0 - g_c)\right| + \sum_{p=1}^{P-1} \left|s(g_p - g_c) - s(g_{p-1} - g_c)\right| \tag{4}$$

According to the definition of 'uniform', there are P+1 'uniform' binary patterns in a circularly symmetric neighbor set of P pixels. Equ. (3) assigns a unique label to each of them, corresponding to the number of '1' bits in the pattern (0,1,...,P), while the 'non-uniform' patterns are grouped under the 'miscellaneous' label (P+1). Thus, the $LBP^{riu2}_{P,R}$ has P+2 distinct output values.

2.2 Structures Extracted by Conventional LBP

The uniform LBP, as a classical version of LBP, has achieved a good performance in texture analysis. Therefore, it is necessary to select the uniform LBP method to analysis the structures that are extracted by LBP. According to the definition of uniform, there are 58 different uniform patterns in (8,R) neighborhood. The pattern '0' means the gray values of P neighbor points smaller than the central pixel's, and corresponds to a plot structure. Patterns that have equal numbers of continuous '0' and '1' correspond to an edge structure. The pattern '255' correlates with two structures. If the gray values of P neighbor points greater than the central pixel's, the pattern expresses a plot structure. And the flat structure is described when the P neighbor points and their central pixel have the same

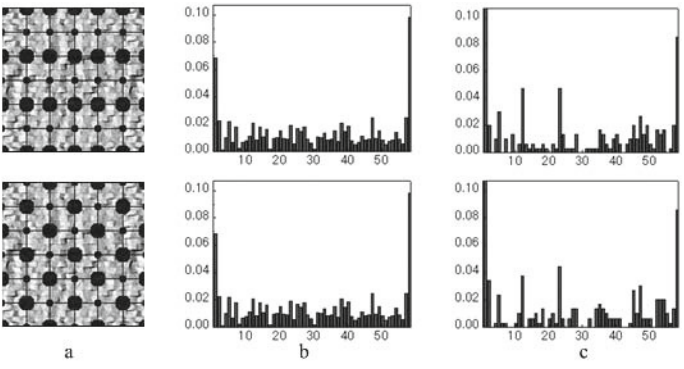


Fig. 1. 1st column: Two texture images have same micro structures. 2nd column: Uniform LBP (P=8, R=1) histograms of left textures. 3nd column: Uniform LBP (P=8, R=1) histograms of the second level of pyramid of left textures. All the histograms are normalized. The Euclidian distances of histograms in subimages (b) and (c) are 0.0029 and 0.0454, respectively.

gray value. The remained uniform patterns are 'L' type corners of the image. These structures are the micro structures of images because the conventional LBP methods only consider a small neighborhood. The non-uniform patterns also correspond to small structures ('Y' corner, 'X' corner, short line and so on).

The performance of conventional LBP is limited, because these methods only consider micro structures of images. The weakness is clear when different texture images have same micro structures. We give an extreme example in Fig. 1. In first column, there are two texture images which have same micro structures but different macro structures. The second column presents uniform LBP (P=8, R=1) histograms of the textures in first column. It's clearly that the uniform LBP worked on orignal images have no contribution to classify the two textures because they have similar LBP feature histograms. The third column gives uniform LBP (P=8, R=1) histograms of the second level of pyramid of texture images in first column. Some differences of the two histograms can be seen in the third colum. The details of image pyramid will be described in next section.

3 Multi-structure Local Binary Pattern

In image field, the isotropic means doing the same operation in every direction, while the anisotropic is just opposite. Studying from the section 2, we can easily find that the basic LBP only considers the isotropic micro structures of images, because it samples with equal space in a small circularly symmetric neighborhood. In this section, we execute the rotation invariant LBP on an image pyramid to extract three different kinds of structures: (1) isotropic micro structures; (2) isotropic macro structures; (3) anisotropic macro structures.

3.1 Image Pyramid

An image pyramid can be created from the original image. We use the sign $I_{l,t}$ to indicate sub-images of an image pyramid. The subscript l stands for the level of the image pyramid and t indicates what template is used to create the sub-image $I_{l,t}$. There are five templates in total. The first template is a 2-dimension Gaussian function $G(x, y, \sigma)$, which is used to smooth image. Referred to SIFT [7], we select the variance σ=1.5.

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{5}$$

Other four anisotropic filters $T_1 \sim T_4$ are used to create anisotropic sub-images of the pyramid. Fig. 2 shows the structures of templates $T_1 \sim T_4$. The templates $T_2 \sim T_4$ are created by rotating the template T_1 clockwise in three different angles (45°, 90° and 135 °).

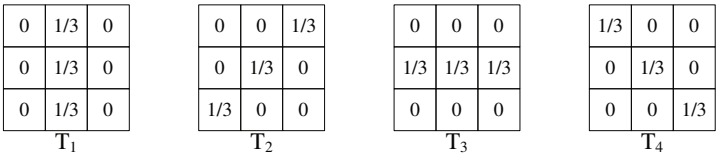


Fig. 2. Four templates ($T_1 \sim T_4$) for drawing anisotropic macro structures

The original image is $I_{0,0}$. The sub-image $I_{l,t}$ (l>0) are created from the image $I_{l-1,0}$ by the following formula:

$$I_{l-1,t} = \begin{cases} (I_{l,0} * G) \downarrow 2, & t = 0 \\ (I_{l,0} * T_t) \downarrow 2, & t = 1, \cdots, 4 \end{cases} \tag{6}$$

where * is the convolution operation, $\downarrow 2$ means downing sample by 2. Here, an approximation is used for extracting anisotropic sub-image by $\downarrow 2$ operation. This approximation is able to reduce the calculations and has little influence when an operation is implemented in small neighborhood. Fig. 3 gives a three-level pyramid for extracting different structures.

3.2 Rotation Invariant Multi-structure Local Binary Pattern

The multi-structure local binary pattern (*Ms-LBP*) can be achieved by running the basic LBP on the image pyramid. The isotropic and anisotropic macro structures are obtained by LBP methods in the sub-images $I_{l,0}$ (l>0) and $I_{l,t}$(l >0,t=1,2,3,4), respectively. The isotropic micro structures are obtained by LBP methods in original image. Similarly, the rotation invariant multi-structure binary pattern can be achieved by carrying out the rotation invariant uniform pattern operator $LBP^{riu2}_{P,R}$ on different pyramid levels. Thus, the sign of our method

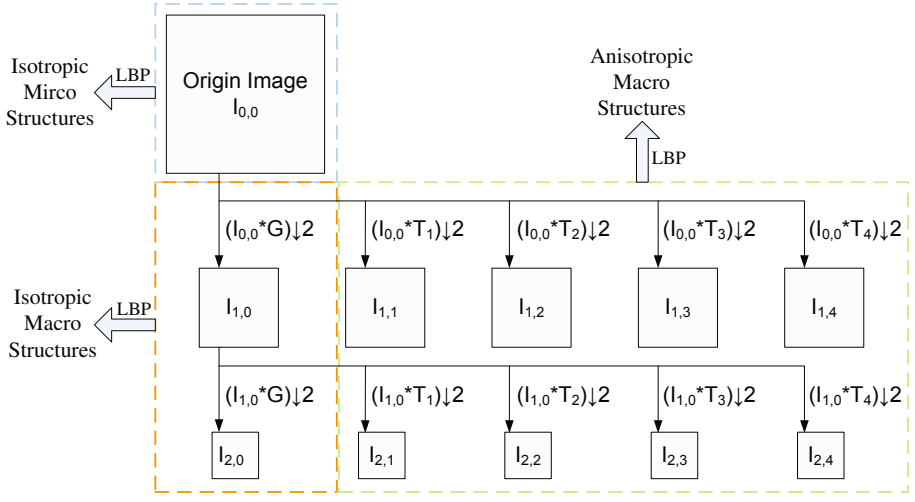


Fig. 3. Extraction of multi-structures in a three-level pyramid

is rewritten as $Ms\text{-}LBP^{riu2}_{P,R}$, where 'P', 'R' and 'riu2' have the same meaning with the operator $LBP^{riu2}_{P,R}$. The final histogram of $Ms\text{-}LBP^{riu2}_{P,R}$ is grouped with $LBP^{riu2}_{P,R}$ histograms of every single sub-images of pyramid:

$$H_{l,t}(k) = \sum_{i=1}^{N}\sum_{j=1}^{M} f(LBP^{riu2}_{l,t,P,R}(i,j),k), k \in [0,K] \tag{7}$$

$$f(x,y) = \begin{cases} 1, & x = y \\ 0, & otherwise \end{cases} \tag{8}$$

where $LBP^{riu2}_{l,t,P,R}(i,j)$ is the $LBP^{riu2}_{P,R}$ value of pixel $I_{l,t}(i,j)$; K is the maximal pattern, $H_{l,t}$ is the $LBP^{riu2}_{P,R}$ histogram of the sub-image $I_{l,t}$; M and N are the sizes of the sub-image of the pyramid.

3.3 Similarity Metric

The similarity of sample and model histograms can be evaluated by a test of goodness-of-fit, which is measured with a nonparametric statistical test. The nonparametric test can avoid making any assumptions about the feature distributions. The log-like distance that is employed by many literatures [9, 14, 17] is a goodness-of-fit statistics and useful for measuring the similar between histograms. The log-like distance between a model M and a sample S is computed as follow:

$$D(S,M) = \sum_{b=1}^{B} S_b \log(M_b) \tag{9}$$

where B is the number of bins and S_b and M_b are, respectively, the values of the sample and model images at the bth bin.

The final similarity distance contains three parts because of the existing of three different kinds of structures. Considering the rotation variance of texture images, we take four anisotropic sub-images $I_{l,t}(l>0, t=1,\cdots,4)$ in one level as a whole to compute the distance. And the procedure is iteratively run four times to find the minimal similarity as the distance of anisotropic macro structures. Comparing with the micro structures, the macro structures located in the pyramid top show little statistical because of the small sizes of sub-images in high levels. Intuitively, the distances of structures on higher level make fewer contributions to classifying samples than the distances of structures on lower level. Thus, the final similarity distance ($D_F(S,M)$) is computed by adding the three groups of distance with different weights in different levels:

$$D_F(S,M) = w_{0,0}D(S_{0,0},M_{0,0}) + \sum_{l=1}^{L} w_{l,0}D(S_{l,0},M_{l,0}) + \sum_{l=1}^{L} w_{l,1}D_{min}(S_l^{an},M_l^{an})$$

$$\begin{cases} D_{min}(S_l^{an},M_l^{an}) = \dfrac{1}{4}\displaystyle\sum_{t=1}^{4} D(S_{l,mod(t+k-1,4)+1},M_{l,t}) \\ k = \arg\min\limits_{j}(\displaystyle\sum_{l=1}^{L}\sum_{t=1}^{4} D(S_{l,mod(t+j-1,4)+1},M_{l,t}), j=0,1,2,3) \end{cases} \tag{10}$$

where $S_{l,t}$ and $M_{l,t}$ stand for sub-images of pyramid of sample S and model M, respectively; w are the distant weights and L is the maximum level of the image pyramid.

The classification rate is a good candidate as the distant weight. There are two parts in every level of the image pyramid except level zero. One part is used for obtaining isotropic macro structures and the other part is used for collecting anisotropic macro structures. But the 0th level of the image-pyramid is an exception, because there is only a sub-image $I_{0,0}$ that is used to extracted isotropic micro structures. We use one part of the pyramid at a time to achieve the task of rotation invariant texture classification. Different parts get different classification rates which correspond to different weights. The result of a log-like distance between two histograms is always a nonpositive value. Therefore, the normalized wrong classification rates are selected as the distance weights.

4 Experimental Results

We demonstrate the performance of our operators on the public texture database, Outex, which is used to study rotation invariant texture classification by many literatures [1,3,14]. We used this database because their texture images are acquired under more varied conditions (viewing angle, orientation and source of illumination) than the widely used Brodatz database. There are 24 classes of textures in Outex database. And these textures are collected under three illuminations and

at nine angles. Our experiments were performed on two test suites of Outex: Outex_TC_00010 (TC10) and Outex_TC_00012 (TC12). TC10 is used for studying rotation invariant texture classification and TC12 is used for researching illuminant and rotation invariant texture classification. The two test suites contain the same 24 classes of textures as shown in Fig. 4. Each texture class is collected under three different illuminants ('inca', 't184' and 'horizon') and nine different angles of rotation (0°, 5°, 10°, 15°, 30°, 45°, 60°, 75° and 90°).

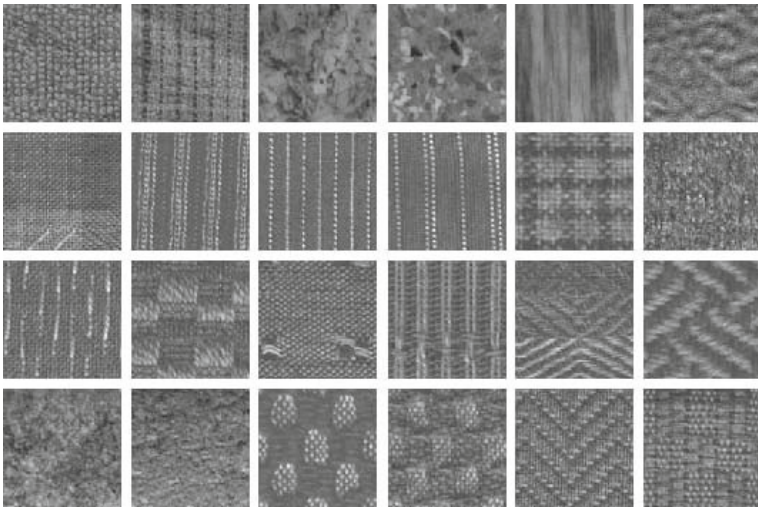


Fig. 4. 128×128 samples of the 24 textures from Outex database

There are 20 non-overlapping 128×128 texture samples for each class under each setting. The experimental setups are as follows:

1. For TC10, classifiers were trained with the reference textures of illuminant 'inca' (20 samples of angle 0° in each texture class), while the 160 (8×20) samples of the other eight rotation angles in each texture class were used for testing the classifier. Hence, there were 480 (24×1×20) models and 3840 (24×8×20) testing samples in total.

2. For TC12, classifiers were trained with the reference textures (20 samples of illuminant 'inca' and angle 0° in each texture class) and tested with all samples captured under illuminant 'tl84' or 'horizon'. Hence, in both of the two illuminant experiments, there are 480 (24×20) models and 4320 (24×20×9) validation samples in total for each illuminant. In order to simplify the name of TC12 test suite, 'TC12t' is short for TC12 'tl84' test suite, and 'TC12h' is short for TC12 'horizon' test suite.

4.1 Calculation of Distance Weights

Both TC12 and TC10 have the same training set that contains 480 samples of illuminant 'inca' in total. There are 20 samples of angle 0° in each texture.

The distance weights were learned on the training set of TC12 and TC10, although the training samples have some differences with the testing samples in angles and illuminants. One sample of each texture was selected as a new training sample for the training classifier at a time, the rest of samples (19×24 samples) were used to test the classifier. We executed the process twenty times, and twenty classification results were given by dividing the samples sets twenty times. The final classification results were the average of the twenty results. In the experiments, the $LBP^{riu2}_{P,R}$ histograms of different parts of pyramid were used to calculate the classification rates. Table 1 presents the right classification rate of different parts of the pyramid in three different sample points P and radius R. The sign '–' in Table 1 presents that no anisotropic structures are extracted in level zero. Five levels are extracted on 128×128 image.

Table 1. Right classification rates (%) with different parts of image pyramid on the training sets of TC10 and TC12

	Isotropic Parts					Anisotropic Parts				
P,R	level0	level1	level2	level3	level4	level0	level1	level2	level3	level4
8,1	76.75	70.78	50.60	38.37	13.44	–	77.21	73.67	48.53	18.43
16,2	81.55	74.44	55.71	33.62	8.66	–	77.98	70.21	38.22	8.00
24,3	81.68	76.79	55.31	13.32	5.04	–	81.88	65.42	15.43	4.67

Results in Table 1 present that the classification rates deteriorate rapidly with the increase of levels, because the sizes of images in high levels are too small to supply enough statistics of structures. Four anisotropic templates cause more anisotropic structures to be extracted than isotropic structures, and give a good performance to the anisotropic parts of image pyramid. As can be seen from Table 1, the classification rates of anisotropic parts are usually higher than the results of isotropic parts in the same levels of image pyramid.

The distance weights were computed by normalizing the wrong classification rates which were obtained by subtracting the correct classification rates from one. Table 2 shows the value of distance weights with different (P, R).

Table 2. Distance weights of different structures

	Isotropic Parts					Anisotropic Parts				
P,R	level0	level1	level2	level3	level4	level0	level1	level2	level3	level4
8,1	0.05	0.07	0.11	0.14	0.20	–	0.05	0.06	0.12	0.19
16,2	0.04	0.06	0.10	0.15	0.20	–	0.05	0.07	0.14	0.20
24,3	0.04	0.05	0.09	0.17	0.19	–	0.04	0.07	0.17	0.19

4.2 Implementation of Multi-structure Local Binary Pattern

The performance of texture classification algorithms is characterized with the percentage of correctly classified samples. The best results of each test suite in experiment are marked in bold font. Five-layered pyramid were used in the experiment according to the size of testing images. Weights in Table 2 were combined with the distances of different parts of the image pyramid to calculate similarity between samples and models. We employed the 3-NN method as a classification principle that has been used by other literatures [9, 14]. The effects of proposed method were compared against six methods: LBP^{riu2}, LBP^{riu2}/VAR, LBP-HF, $LBPV^{u2}_{P,R}GM_{ES}$, $DLBP$ and $MR8$. The method LBP^{riu2} is a useful rotation invariant method. And combining with local variance (VAR), LBP^{riu2}/VAR obtains a good performance. LBP histogram Fourier features (LBP-HF), LBP variance with global matching ($LBPV^{u2}_{P,R}GM_{ES}$) and dominant local binary patterns ($DLBP$) are improved versions of basic LBP. For comparing, we gave the results of $DLBP$ under best parameters ($DLBP_{R=3} + NGF$) in Table 3. $MR8$ is a state-of-the-art statistical algorithm.

Table 3 shows the advantages of the proposed method. The high scores 99.30%, 98.26% and 97.08% are obtained by the operator $Ms\text{-}LBP^{riu2}_{16,2}$ on three different suits(TC10, TC12t and TC12h), respectively. The superiority of our method is obvious on the test suits TC12 which contains both illuminant and rotation variant of textures. Textures under different illuminant usually have different micro structures, but their macro structures are very similar. Thus, compared with other operators, our method works well on testing sets TC12. It could find that the $Ms\text{-}LBP^{riu2}_{P,R}$ method usually excel its counterparts under the same parameters (P,R) and in the same testing sets. This situation is particularly clear when the parameter (P,R) equals to (8,1) because macro structures are also extracted in $Ms\text{-}LBP^{riu2}_{8,1}$. The best results of our method are obtained with parameter (16, 2). And the performance degrades a little with parameter (24,3). The phenomenon is distinct in Table 1, especially the results on large levels of image pyramid. Because the sizes of sub-image in high level of image pyramid are very small, the total numbers of feature is very few. At the same time, the dimension of histogram increases with the number of sampling points P. It's known that a high-dimensional histogram with few features is not enough to describe the distribution of features in the statistical sense. And the operator $Ms\text{-}LBP^{riu2}_{24,3}$ belongs to this situation. As can be seen from the Table 3, the performance of $Ms\text{-}LBP^{riu2}_{P,R}$ degrades a little with large sample points P=24, because the high levels of image pyramid supply few feature with large number of bins of the histogram.

Although good results are obtained, our method needs more time to classify a texture than most LBP operators. We select the best parameter (P,R)=(16,2) and execute these operators on a computer with the Intel CPU 2.8GHz. Our method needs 0.466s to classify a texture, while $LBP^{riu2}_{P,R}$ only needs 0.012s. The classification time of our method is less than the MR8 operator (2.257s), because the MR8 needs to find 8 maximum responses after 38 filters convoluting with

Table 3. Correct Classification rate (%) for TC10 and TC12 using different methods

P,R	8,1			16,2			24,3		
	TC10	TC12t	TC12h	TC10	TC12t	TC12h	TC10	TC12t	TC12h
$LBP^{riu2}_{P,R}$	83.31	69.86	62.94	92.29	86.25	83.61	96.38	89.81	88.75
$LBP^{riu2}_{P,R}/VAR_{P,R}$	95.81	78.73	77.27	97.97	87.06	85.90	97.48	86.81	87.27
$LBP\text{-}HF_{P,R}$	83.26	76.20	78.45	93.93	88.15	86.46	97.97	91.50	87.66
$LBPV^{u2}_{P,R}GM_{ES}$	73.64	72.47	76.57	93.90	90.25	94.28	97.76	95.39	95.57
$MS\text{-}BP^{u2}_{P,R}$	97.87	94.98	91.76	**99.30**	**98.26**	**97.08**	98.26	96.46	94.72
$MR8$	92.5(TC10), 90.9(TC12t), 91.1(TC12h)								
$DLBP_{R=3}+NGF$	99.1(TC10), 93.2(TC12t), 90.4(TC12h)								

the image and compare very 8-dimension vector in an image with all the textons to build histograms.

5 Conclusions

The conventional LBP methods only focus on micro structures of images, although they have already been powerful in texture analysis. In this paper, we executed the rotation invariant uniform LBP on the image pyramid to extract three different structures (isotropic micro structures, isotropic macro structures and anisotropic macro structures). The experiment results on Outex database demonstrate the advantages of our method. The performance of proposed method is limited by the size of images, because small images are not enough to supply large macro structures. Fortunately, the texture images are different from other images, due to they are full of repeat patterns. So in the future, some texture synthesis methods could be used to create large size texture image. And more stable multi-structure local binary patterns could be achieved on the synthesized texture images.

Acknowledgement. This work was supported by the Chinese National 863 Grand No. 2009AA12Z109.

References

1. Ahonen, T., Matas, J., He, C., Pietikäinen, M.: Rotation invariant image description with local binary pattern histogram fourier features. In: Salberg, A.-B., Hardeberg, J.Y., Jenssen, R. (eds.) SCIA 2009. LNCS, vol. 5575, pp. 61–70. Springer, Heidelberg (2009)
2. Davis, L., Johns, S., Aggarwal, J.: Texture analysis using generalized co-occurrence matrices. IEEE Transactions on Pattern Analysis and Machine Intelligence 1, 251–259 (1979)

3. Guo, Z., Zhang, L., Zhang, D.: Rotation invariant texture classification using lbp variance (lbpv) with global matching. Pattern Recognition 43, 706–719 (2009)
4. Heeger, D.J., Bergen, J.R.: Pyramid-based texture analysis/synthesis. In: Proceedings of SIGGRAPH 1995, pp. 229–238 (1995)
5. Laine, A., Fan, J.: Texture classification by wavelet packet signatures. IEEE Transactions on Pattern Analysis and Machine Intelligence 15, 1186–1191 (1993)
6. Liao, S., Law, M., Chung, A.: Dominant local binary patterns for texture classification. IEEE Transactions on Image Processing 18, 1107–1118 (2009)
7. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60, 91–110 (2004)
8. Mäenpää, T., Ojala, T., Pietikäinen, M., Soriano, M.: Robust texture classification by subsets of local binary patterns. In: Proc. 15th International Conference on Pattern Recognition, Barcelona, Spain, pp. 947–950 (2000)
9. Mäenpää, T., Pietikäinen, M.: Multi-scale binary patterns for texture analysis. In: Bigun, J., Gustavsson, T. (eds.) SCIA 2003. LNCS, vol. 2749, pp. 885–892. Springer, Heidelberg (2003)
10. Mäenpää, T., Pietikäinen, M.: Texture analysis with local binary patterns. In: Handbook of Pattern Recognition and Computer Vision, 3rd edn., pp. 197–216. World Scientific, Singapore (2005)
11. Manjunath, B.S., Ma, W.Y.: Texture features for browsing and retrieval of image data. IEEE Transactions on Pattern Analysis and Machine Intelligence 18, 837–842 (1996)
12. Ojala, T., Pietikäinen, M., Harwood, D.: A comparative study of texture measures with classification based on featured distributions. Pattern Recognition 29, 51–59 (1996)
13. Ojala, T., Valkealahti, K., Oja, E., Pietikäinen, M.: Texture discrimination with multidimensional distributions of signed gray-level differences. Pattern Recognition 34, 727–739 (2001)
14. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 971–987 (2002)
15. Randen, T., Husoy, J.: Filtering for texture classification: A comparative study. IEEE Transactions on Pattern Analysis and Machine Intelligence 21, 291–310 (1999)
16. Varma, M., Zisserman, A.: A statistical approach to texture classification from single images. International Journal of Computer Vision 62, 61–81 (2005)
17. Zhao, G., Pietikäinen, M.: Dynamic texture recognition using volume local binary patterns. In: Vidal, R., Heyden, A., Ma, Y. (eds.) WDV 2005/2006. LNCS, vol. 4358, pp. 165–177. Springer, Heidelberg (2007)

Unsupervised Moving Object Detection with On-line Generalized Hough Transform

Jie Xu, Yang Wang, Wei Wang, Jun Yang, and Zhidong Li

National ICT Australia
University of New South Wales
{jie.xu,yang.wang,jun.yang,zhidong.li}@nicta.com.au,
weiw@cse.unsw.edu.au

Abstract. Generalized Hough Transform-based methods have been successfully applied to object detection. Such methods have the following disadvantages: (i) manual labeling of training data ; (ii) the off-line construction of codebook. To overcome these limitations, we propose an unsupervised moving object detection algorithm with on-line Generalized Hough Transform. Our contributions are two-fold: (i) an unsupervised training data selection algorithm based on Multiple Instance Learning (MIL); (ii) an on-line Extremely Randomized Trees construction algorithm for on-line codebook adaptation. We evaluate the proposed algorithm on three video datasets. The experimental results show that the proposed algorithm achieves comparable performance to the supervised detection method with manual labeling. They also show that the proposed algorithm outperforms the previously proposed unsupervised learning algorithm.

1 Introduction

The detection of moving objects in videos, especially pedestrians or vehicles, is an important task in many vision applications, such as video compression, video surveillance, and content-based video retrieval. Numerous approaches have been proposed in the literature for object detection. Currently the predominant approach for object detection is the sliding window approach [1], [2], in which a learned classifier examines the image features over locations and scales to predict the presence of objects in subwindows. Though it has been demonstrated effective in many cases, it can be easily affected by background clutters and occlusions. To cope with the occlusion problem, part-based approaches [3], [4] which model objects as collections of parts are proposed.

The Generalized Hough Transform based methods [5], [6] can be categorized as part-based approaches. Each of them requires a class-specific codebook to cast probabilistic votes for object hypotheses. The codebook can be generated using generative clustering methods [5], and discriminative clustering methods [6]. Each cluster centroid corresponds to one codebook instance. At runtime, feature descriptors from the testing data are matched against the codebook instances, and valid matches then cast probabilistic votes for object hypotheses.

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 145–156, 2011.

The additive nature of Generalized Hough Transform makes the detector robust to partial occlusions. However, these methods have the following disadvantages: (i) manual labeling of training data is required for the codebook construction; (ii) the codebook is constructed in an off-line manner, which cannot adapt to new data after the construction ends.

Several approaches have been proposed to tackle the problem of manual labeling of training data. The idea of *co-training* is proposed to incrementally generate a large amount of labeled data automatically from a small manually labeled set [7]. Given a small hand labeled set, a *pair* of classifiers are trained on two independent "views" of the data [7]. Co-training then generates the additional training data from the unlabeled data, by using each classifier's prediction to enlarge the other classifier's training set [8]. Alternatively, Wu *et al.* uses a small labeled training set to train an automatic labeler, which is then used to collect the training samples for the on-line boosting in [9]. Both approaches require hand labeled sets for initialization. To overcome the limitation of hand labeling, the idea of automatic labeling is proposed. Nair *et al* employs the motion based object detector as the labeler in [10]. However, motion based object detector is not robust, and can be affected by shadows, reflections and illumination changes. To improve such labeler, Roth *et al.* uses the PCA-based reconstructive model [11], to verify the motion detection results. As for the codebook construction for the Generalized Hough Transform, tree-based codebooks have become popular recently. The Extremely Randomized Trees [6], and the Random Forests [12] have been demonstrated to improve the performance of the Generalized Hough Transform. Such trees are usually learned offline, however Saffari *et al.* recently propose an on-line algorithm to enable the on-line learning of Random Forests [13].

In this paper, we propose an unsupervised moving object detection algorithm, with on-line Generalized Hough Transform. Our contributions are two-fold: (i) an unsupervised on-line training data selection algorithm based on Multiple Instance Learning (MIL); (ii) an on-line Extremely Randomized Trees construction algorithm for on-line codebook adaptation. The most related algorithm to our automatic training data selection algorithm is the co-training algorithm [7], and also the conservative learning algorithm [11]. Unlike the co-training algorithm, our algorithm does not require any hand labeling. In the conservative learning algorithm, a reconstructive model is employed to verify the motion detection results. Only the sufficiently consistent motion detections would be used to build the reconstructive model, and hence it might result in a biased training set. In contrast, our algorithm employs an instance selection scheme to produce a training set with less selection bias. For our proposed on-line Extremely Randomized Trees algorithm, the most related work is the on-line Random Forest algorithm by Saffari *et al.* in [13]. Different from the on-line Random Forest, our on-line Extremely Randomized Trees do not require the bootstrapping, and hence it is more computationally efficient.

The rest of the paper is organized as follows: the proposed work is described in Section 2, followed by the experimental results in Section 3. Our final conclusions are presented in Section 4.

2 Proposed Work

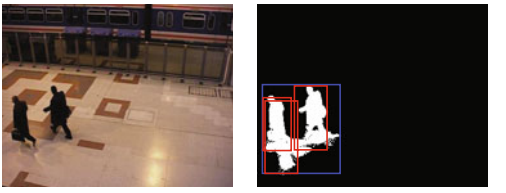
In this section, we present our unsupervised moving object detection algorithm with on-line Generalized Hough Transform. We design our automatic labeler based on Multiple Instance Learning for training sample selection. Given a set of noisy detection results from the background subtraction, the automatic labeler selects training samples automatically and unbiasedly. The on-line learning algorithm then uses the selected samples for codebook adaptation.

2.1 Automatic On-line Instance Selection

We present our automatic labeler design in this section. An automatic labeler is actually an object detector, which selects sub-windows that contain objects. Generally speaking, there are two issues in the design of a labeler for object detection. One issue is the labeler's error, which can be categorized as the alignment error and the labeling error. An alignment error occurs when the sub-window selected by the labeler contains an object with inaccurate size of positions, whereas a labeling error occurs when the selected sub-window contains no object. The other issue is the labeler's bias. The labeler should not introduce any bias into the produced training data, otherwise it may mislead the detector. For instance, if the labeler systematically fails to collect some certain type of training sample, the detector would not be able to recognize the corresponding object. We will show how our design can cope with these two issues.

We begin our design with background subtraction. Given a video, background subtraction generates a set of foreground blobs, which comprise the training samples for selection. Since background subtraction is not robust against environmental factors, alignment and labeling error might occur. To handle the errors, we introduce the Multiple Instance Learning (MIL) to our labeler design. In MIL, the training data comes in the form of "bags", where all the instances in one bag share a label. A positive bag means it contains at least one positive instance, whereas a negative bag means all the instances are negative. The advantage of MIL is that, it can handle both the ambiguity and noises in the instance labeling. In our problem, each foreground blob corresponds to a positive bag. Given one foreground blob, in order to locate the possible locations of individual persons, a smoothed histogram of foreground heights over the x-axis is computed. We assume that the tops of objects correspond to the peaks of the histogram. After the peaks are located, we crop the corresponding instances using bounding boxes. Figure 1a depicts the image frame, and Figure 1b demonstrates the detected foreground blob and its bag formulation. In Figure 1b, the blue rectangle is the foreground blob, while the red rectangles correspond to the instances inside the bag. As shown in Figure 1b, the foreground blobs contains two pedestrians, but there are more than two instances found in the corresponding bag due to the noisy motion detection result.

To deal with the noisy detection, we propose to use the following scheme to select instances from all the bags. Let $\mathbb{B} = \{B_1^+, B_2^+, ..., B_n^+\}$ be the set of all positive bags. The goal of the selection is to select the instance B_{gh} that has high confidence $\text{Conf}(B_{gh})$, which is defined in the follows:



(a) (b)

Fig. 1. The formation of a positive bag. (a) The image frame. (b) The detected foreground and its bag formulation. The blue rectangle is the foreground blob, while the red rectangles correspond to the instances inside the corresponding bag. This figure is best viewed in color mode.

$$Conf(B_{gh}) = \prod_{k} Pr(B_{gh}|B_k^+),\ 1 \leq k \leq n,\ k \neq g, \tag{1}$$

where $Pr(B_{gh}|B_k^+)$ is estimated based on the Noisy-OR model [14]:

$$Pr(B_{gh}|B_k^+) \propto \{1 - \prod_{j}[1 - Pr(B_{gh}|B_{kj}^+)]\}. \tag{2}$$

We can design different estimations for $Pr(B_{gh}|B_{kj}^+)$ based on different data. For the task of object detection, we intend to evaluate the similarity between two object blob silhouettes, and $Pr(B_{gh}|B_{kj}^+)$ is estimated as $Pr(B_{gh}|B_{kj}^+) \propto \exp\{-D(B_{gh}, B_{kj}^+)\}$, where $D(B_{gh}, B_{kj}^+)$ measures the distance between the silhouettes B_{gh} and B_{kj}^+. In this paper, we design the distance based on distance transform. To make sure our method is as general as possible, only positive bags are required for the computation of the evidence. In the situations when the negative bags are also available, we can use the Evidence Confidence metric proposed in [15].

The aforementioned instance selection scheme is a batch process. To enable the on-line selection, we propose an on-line algorithm for the above selection algorithm. We choose to realize the on-line learning by selecting the instances from every R frames, where R is a pre-defined value to determine the size of the interval. The proposed on-line learning algorithm is presented in Algorithm 1.

2.2 On-line Extremely Randomized Trees

Given a set of selected instances, we attach shape context descriptors to the sampled points from the corresponding silhouettes. The obtained descriptors are used to construct a codebook of shapes for object silhouettes. The codebooks for the Generalized Hough Transform are usually generated using unsupervised k-means clustering [5], [16]. We call them generative codebooks as there is no discrimination involved. Recently discriminative codebook generation methods are proposed [12], [6]. The generated codebooks are considered as discriminative codebooks as

Algorithm 1. Automatic On-line Instance Selection

INPUTS:
$\mathbb{F}_t$ - The extracted foreground from frame t to frame $t+R$, where R is a pre-defined value
K - The number of instances to select from all the instances
OUTPUTS:
A set of instances B_{ij} for training the randomized trees
1: Form the positive bags $\mathbb{B} = \{B_1^+, B_2^+, ..., B_n^+\}$ based on $\mathbb{F}_t$
2: Compute the confidence for all the instances B_{ij}
3: Select the top K instances B_{ij} with the highest confidence

they are trained in a supervised way. The supervision enables the codebook entries to cast more reliable probabilistic votes. In [12], a Random Hough Forest is constructed using both positive and negative image patches, with an objective function that measures the class and offset uncertainty. On the other hand, a set of Extremely Randomized Trees are constructed in [6], and the trees are grown using an objective function that combines the discrimination and regression. The discriminative codebooks are shown to outperform the generative codebook in the experiments. As a result, we use the discriminative codebook in our paper.

We choose the Extremely Randomized Tress [17] as our discriminative codebook. The randomized trees algorithm [17] constructs an ensemble of decision or regression trees. And each tree is grown by splitting each node into two child nodes, using the random split that achieves the best decision or regression performance based on the whole training set. The randomized trees algorithm is firstly proposed for classification, and Okada employs it as the codebook for the Hough voting [6]. Each primitive image feature passes through each randomized tree until it reaches one of the leaf node. The leaf node contains information about the discrimination of the image feature(whether it belongs to an object or not), and possible object locations are collected during the training. The response of one image feature is an ensemble of the responses from all the trees. Using the responses, each feature can cast probabilistic votes for object hypotheses. The randomized tree construction algorithms in [17], [6] are all based on the whole training set. It is not appropriate to use them under our problem settings, as we want to be able to update the trees in an on-line fashion. Inspired by the on-line random forest algorithm in [13], we propose an on-line learning algorithm for constructing the randomized trees here. It is noted that the randomized trees are different from random forests as there is no bootstrapping involved in the randomized trees [17].

We build each randomized tree as a decision tree, which contains the decision nodes and the leaf nodes. Unlike the leaf node, each decision node retains no object location but only a split condition $s = \{f_d, \theta_d\}$, where f_d and θ_d are a randomly chosen attribute from the image feature vector, and its threshold respectively. The split s is the best split chosen from a set of random splits

$S = \{s_1, s_2, ..., s_N\}$ based on some quality measure. In our paper, the information gain is chosen as the quality measure. Denote M as the set of image features in the current node. Let M_L and M_R be the images features in the left child node and right child node respectively, according to the split s. The information gain of split s is $IG_s(M) = \frac{|M_L|}{|M|}E(M_L) + \frac{|M_R|}{|M|}E(M_R) - E(M)$, where $E(M) = -\sum_{i=1}^{C} p_i \log(p_i)$ is the entropy for C classes.

When in off-line mode, all the data is available, and therefore a robust estimate can be made at each decision node. In the on-line mode, however, the data is gathered over time, and hence, when to split depends on the following factors: i) whether there is enough data for the robust estimate of statistics; ii) whether the split is good enough in terms of the quality measure. Based on the these factors, two hyper-parameters are introduced for the on-line learning of a random tree: i) the minimum number of training data (i.e., shape context descriptors) γ to gather before making a split; ii) the minimum information gain δ for a node to split. And therefore, a node can be split into two child nodes only if $|M| > \gamma$ and $\exists s \in S,\ IG_s(M) > \delta$. The values of γ and δ are set in a similarly way to method mentioned in [13].

The on-line learning algorithm for the randomized trees construction is presented in Algorithm 2. The input to algorithm is either a positive or negative training sample $\langle\ x,\ y\ \rangle$, which contains a feature descriptor x and its label $y \in \{1, 0\}$. In this paper we use the shape context as the feature descriptor. The positive feature descriptors describe the sampled points from the selected instance B_{ij} from Algorithm 1, whereas the negative descriptors describe the

Algorithm 2. On-line Extremely Randomized Trees

INPUTS:
$\langle\ x,\ y\ \rangle$ - a training sample from a sampled keypoint
γ - The minimum number of training data to gather before making a split
δ - The minimum information gain for a node to split
T=$\{t_1, t_2, ..., t_n\}$ - A set of Extremely Randomized Trees
OUTPUTS:
T'=$\{t'_1, t'_2, ..., t'_n\}$ - The updated Extremely Randomized Trees

```
for Each Extremely Randomized Tree t_i do
    l_j ← locateLeaf(x, t_i)
    l_j ← appendData( l_j, ⟨x, y⟩)
    if |l_j| > γ then
        S ← createSplts( l_j)
        if ∃s ∈ S, IG_s(l_j) > δ then
            createLeftChild( l_j, s)
            createRightChild( l_j, s)
        end if
    end if
end for
```

sample points from the background edges. Positive samples also retain the offsets to the centroids of an object, so that the constructed randomized tree can be used for probabilistic voting, which is detailed in Section 2.3. When updating a tree, a training sample firstly passes each randomized tree until it reaches the leaf node. After appending the new feature to the leaf node, we calculate whether it is necessary to split the current leaf. In the case of a split, the data retained in the old leaf node will be propagated to its child nodes, and the old leaf node becomes a decision node.

2.3 Object Detection

We begin the moving object detection with identifying moving edges between adjacent frames. We apply Canny edge detection [18] to obtain the edge map for each frame. Moving edges are then extracted by comparing edges between adjacent frames. We then sample keypoints from the moving edges, and attach shape context descriptor to each sampled keypoints. Let $F = \{f_1, f_2, ...f_n\}$ be the shape context descriptors obtained from the current frame, F will be then fed into the randomized trees $T = \{t_1, t_2, ..., t_n\}$ to cast probabilistic votes for an object o and its location x. The probabilistic vote $p(o, x|f_i, T)$ from feature f_i can be decomposed as $p(o|f_i, T)p(x|o, f_i, T)$. The first term $p(o|f_i, T)$ is a probabilistic output from the ensemble of trees. Denote M_{f_i,t_j} as the set of training features belong to the leaf node to which f_i reaches in tree t_j. Let the number of training features in M_{f_i,t_j} be $N_{f_i,t_j} = |M_{f_i,t_j}|$, and that of the positive features be $N^p_{f_i,t_j} = |M^p_{f_i,t_j}|$. The purity of the leaf node can be defined as $\gamma_{f_i,t_j} = \frac{N^p_{f_i,t_j}}{N_{f_i,t_j}}$. We only consider the trees with leaf nodes whose purity is higher than a predefined threshold. Assume the number of such trees to be $N^o_{f_i}$, and $p(o|f_i, T)$ is defined as $p(o|f_i, T) = \frac{N^o_{f_i}}{N_T}$, where N_T is the number of randomized trees.

Algorithm 3. Moving Object Detection with On-line Generalized Hough Transform

AUTOMATIC LABELING AND ON-LINE LEARNING
for every R frames **do**
  Perform background subtraction, and group foreground pixels into blobs
  Use the Algorithm 1 to select instances from the foreground blobs
  Attach descriptors to sample edge points from instances and background
  Use the descriptors to update the randomized trees based on Algorithm 2
end for
ON-LINE MOVING OBJECT DETECTION
for Each frame **do**
  Identify moving edges
  Attach descriptors to the sample edge points from the moving edges
  Use the randomized trees to cast probabilistic votes based on the descriptors
end for

The second term $p(x|o, f_i, T)$ describes the distribution of possible object centroid location in regard to f_i supposing f_i being part of the object. The distribution is estimated using a non-parametric density estimation using all the trees:

$$p(x|o, f_i, T) \propto \sum_{j=1}^{N_T} \{\gamma_{f_i,t_j} \sum_{k \in M^p_{f_i,t_j}} K(\frac{x - x^p_k(f_i)}{b(x^p_k)})\}, \quad (3)$$

where $K(.)$ is a window function, $b(.)$ is its bandwidth, and $x^p_k(f_i)$ corresponds to the object centroid location relative to the feature f_i based on the positive training feature x^p_k.

The complete unsupervised moving object detection algorithm is summarized in Algorithm 3. The proposed algorithm updates the randomized trees using the collected training samples from every R frames, and then the updated trees are used to cast probabilistic votes for object hypotheses based on the moving edge detection results.

3 Experiments

Experimental Setup. We evaluate the performance of the proposed framework on moving object detection using three video datasets. The first two of them, including the PETS2006 benchmark set[1] and the i-LIDS dataset[2], are indoor video surveillance on pedestrian activities. The third dataset contains outdoor traffic surveillance video captured in a highway during daytime.

Evaluation Metric. We follow the evaluation criteria employed in [5] that covers three categories, and they are *relative distance*, *cover*, and *overlap*. The *relative distance* measures the distance between the center of a bounding box and that of the ground truth. The *cover* and *overlap* measure how much area of the ground truth bounding box is covered by the detection hypothesis, and vice versa. A hypothesis is classified as a true positive if the *relative distance* ≤ 0.5 and both *cover* and *overlap* are above 50%.

3.1 The PETS2006 Dataset

We evaluate the two components of the proposed framework using the PETS2006 dataset. We extract four sequences from the dataset, and use one sequence for training, and the rest three for testing. The number of moving objects in the testing sequences are 842, 312 and 413 respectively.

The Automatic On-line Instance Selection. To evaluate our automatic labeler, we collect two training sets from the training sequence using manual selection and the proposed labeler respectively. We then use them to train two sets of batch Extremely Randomized Trees [6] respectively. The obtained randomized trees are tested on the testing sequences based on the detection algorithm

[1] http://www.cvg.rdg.ac.uk/PETS2006/data.html
[2] http://www.elec.qmul.ac.uk/staffinfo/andrea/avss2007.html

detailed on Section 2.3. The precision recall curves of both sets of trees are depicted in Figure 2a. As shown in the figure, the randomized trees trained on the automatic labeled set achieve comparable performance with the trees trained on the hand labeled set. It is noted that, the former even outperforms the latter on the third testing sequence. This might be due to the selection bias of the manual labeling. Sample detection results can be found in Figure 3.

The On-line Extremely Randomized Trees. We compare the proposed on-line learning algorithm for the randomized trees with the corresponding batch learning algorithm [6]. Given the same training set, we construct two sets of randomized trees using the on-line and batch learning algorithm respectively. These two sets of randomized trees are then tested on the testing sequences. Figure 2b depicts the precision recall curves of both sets of randomized trees on the testing sequences. It can be seen from the curves that, the on-line learning algorithm reaches comparable or even better precision than the batch learning algorithm at the same recall value. These indicate that the proposed on-line learning algorithm for the randomized trees adapts to the incoming data better than the batch learning algorithm.

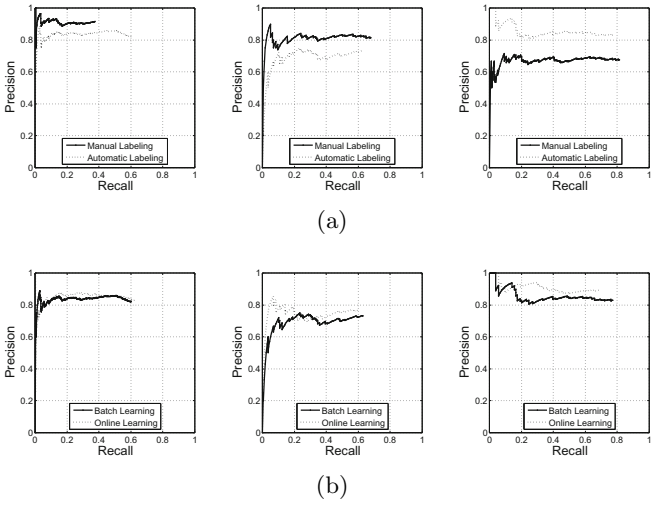


Fig. 2. The precision recall curves on the PETS2006 dataset: (a) the curves correspond to manual and automatic labeling, (b) the curves correspond to batch and on-line learning

3.2 The i-LIDS Dataset

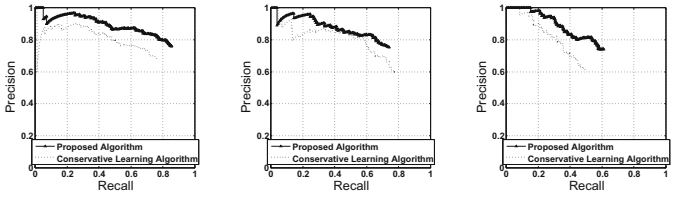
As our second experiment, we compare the proposed algorithm with the unsupervised on-line conservative learning algorithm in [11]. The labelers of both algorithms are based on the simple background subtraction results from the training video. In [11], a reconstructive model based on appearance and shape is employed to verify the foreground blobs. In this experiment, only the shape

(a) Sequence 1 (b) Sequence 2 (c) Sequence 3

Fig. 3. Sample detection results of randomized trees based on manual and automatic labeling. For each pair of results, the manual labeling-based detection is shown on the left, and the automatic labeling-based detection is shown on the right.

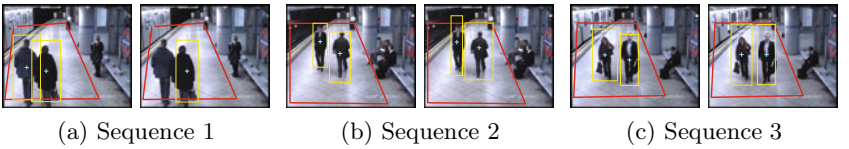
information is used for a fair comparison of both frameworks. The conservative labeler only considers the foreground blobs whose aspect ratio are within the predefined limits. For instance, Figure 1 depicts one frame in the training set, and also the corresponding foreground blob. The aspect ratio of the blob exceeds the predefined limit, and hence is not considered by the conservative learning. The conservative learning algorithm might fail to capture the multi-modal nature of the data due to its conservativeness. On the other hand, our proposed labeler does not have such requirement, and also accepts this blob for instance selection. As a result, the proposed algorithm would have less selection bias. For the object detection, we use the proposed on-line randomized trees for object detector for both algorithms.

We extract three sequences from the i-LIDS dataset, and each of them contains 250, 202, and 262 objects respectively. We compare both algorithms on these sequences, and their performances are shown in the precision recall Curves



(a) Sequence 1 (b) Sequence 2 (c) Sequence 3

Fig. 4. The performance of the proposed algorithm and the conservative learning algorithm on the i-LIDS dataset



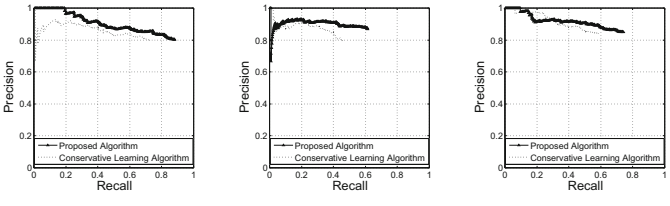
(a) Sequence 1 (b) Sequence 2 (c) Sequence 3

Fig. 5. Sample detection results on the i-LIDS dataset. For each pair of results, the detection obtained by the proposed algorithm is shown in the left, and that obtained by the conservative learning is shown on the right.

in Figure 4. It can been observed from the figure that, the proposed algorithm outperforms the conservative learning framework. Our framework reaches higher precision in all testing sequences. This indicates that the proposed framework captures the multi-modal nature of pedestrian silhouettes better than the conservative framework. Sample detection results can be found in Figure 7.

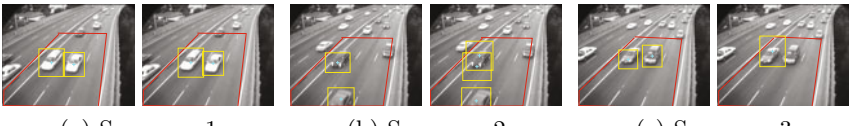
3.3 The Traffic Dataset

As our last experiment, we compare the proposed algorithm with the unsupervised on-line conservative learning algorithm in [11] for vehicle detection. Similarly, only shape information is used here, and we also use the on-line randomized trees for object detection. The performance of both learning algorithms are shown in Figure 6. It is seen in the figure that, the proposed algorithm slightly outperforms the conservative learning algorithm. This result indicates that the silhouettes of the vehicles might follow a unimodal distribution, since most vehicles in the videos are vans and trucks. Sample detection results can be found in Figure 7.



(a) Sequence 1 (b) Sequence 2 (c) Sequence 3

Fig. 6. The performance of the proposed algorithm and the conservative learning algorithm on the traffic set



(a) Sequence 1 (b) Sequence 2 (c) Sequence 3

Fig. 7. Sample detection results on the traffic dataset. For each pair of results, the detection made by the proposed algorithm is shown in the left, and that made by the conservative learning algorithm is shown on the right.

4 Conclusions

We have presented a novel algorithm for on-line unsupervised learning of object detection system. The basic idea is to start with a simple motion detection system, and then select the optimal foreground blobs based on the Multiple Instance Learning. Subsequently the selected blobs are used to construct a set

of Extremely Randomized Trees in an on-line manner. We have evaluated the algorithm on three video datasets. The experimental results demonstrate that our algorithm outperforms the on-line conservative learning algorithm.

Acknowledgements. National ICT Australia (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 Center of Excellence program. Dr. Wei Wang is supported by ARC Discovery Grant DP0987273.

References

1. Dalai, N., Triggs, B., Rhone-Alps, I., Montbonnot, F.: Histograms of oriented gradients for human detection. In: CVPR, vol. 1 (2005)
2. Munder, S., Gavrila, D.: An experimental study on pedestrian classification. TPAMI (2006)
3. Andriluka, M., Roth, S., Schiele, B.: People-tracking-by-detection and people-detection-by-tracking. In: CVPR (2008)
4. Fergus, R., Perona, P., Zisserman, A.: Weakly supervised scale-invariant learning of models for visual recognition. IJCV (2007)
5. Leibe, B., Leonardis, A., Schiele, B.: Robust Object Detection with Interleaved Categorization and Segmentation. IJCV (2008)
6. Okada, R.: Discriminative Generalized Hough Transform for Object Detection. In: ICCV (2009)
7. Balcan, M., Blum, A., Yang, K.: Co-training and expansion: Towards bridging theory and practice. In: NIPS (2005)
8. Javed, O., Ali, S., Shah, M.: Online detection and classification of moving objects using progressively improving detectors. In: CVPR (2005)
9. Wu, B., Nevatia, R.: Improving part based object detection by unsupervised, online boosting. In: CVPR (2007)
10. Nair, V., Clark, J.: An unsupervised, online learning framework for moving object detection. In: CVPR (2004)
11. Roth, P., Grabner, H., Skocaj, D., Bischof, H., Leonardis, A.: On-line conservative learning for person detection. In: VS-PETS (2005)
12. Gall, J., Lempitsky, V.: Class-Specific Hough Forests for Object Detection. In: CVPR (2009)
13. Saffari, A., Leistner, C., Santner, J., Godec, M., Bischof, H.: On-line Random Forests. In: The 3rd On-line learning for Computer Vision Workshop (2009)
14. Maron, O., Lozano-Pérez, T.: A framework for multiple-instance learning. In: NIPS (1998)
15. Li, W., Yeung, D.: Localized content-based image retrieval through evidence region identification. In: CVPR (2009)
16. Maji, S., Malik, J.: Object detection using a max-margin hough transform. In: ICCV (2009)
17. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Machine Learning (2006)
18. Canny, J.: A computational approach to edge detection. TPAMI (1986)

Interactive Event Search through Transfer Learning

Antony Lam[1], Amit K. Roy-Chowdhury[2], and Christian R. Shelton[1]

[1] Dept. of Computer Science & Engineering, University of California, Riverside
{antonylam,cshelton}@cs.ucr.edu
[2] Dept. of Electrical Engineering, University of California, Riverside
amitrc@ee.ucr.edu

Abstract. Activity videos are widespread on the Internet but current video search is limited to text tags due to limitations in recognition systems. One of the main reasons for this limitation is the wide variety of activities users could query. Thus codifying knowledge for all queries becomes problematic. Relevance Feedback (RF) is a retrieval framework that addresses this issue via interactive feedback with the user during the search session. An added benefit is that RF can also learn the subjective component of a user's search preferences. However for good retrieval performance, RF may require a large amount of user feedback for activity search. We address this issue by introducing Transfer Learning (TL) into RF. With TL, we can use auxiliary data from known classification problems different from the user's target query to decrease the needed amount of user feedback. We address key issues in integrating RF and TL and demonstrate improved performance on the challenging YouTube Action Dataset[1].

1 Introduction

The growth of video sharing websites has resulted in a wealth of Internet videos (mostly of activities) available to users. Automated search of these videos present interesting challenges as the number of activities is arbitrarily large. In addition to the high variability of activities themselves, Internet videos typically exhibit greater variability in quality, camera movement, and lighting when compared with those of TV programs such as news broadcasts. Thus retrieval of such videos is still largely limited to the use of associated text tags.

However, search based on only text is limiting so direct analysis of video content is still desirable. The problem is that users could query for a vast array of activities and it would be very difficult to train high-level semantics for every possible query. In addition, if a user query were subjective (e.g. what the user thinks are "nice basketball shots"), there would be no way to train a system a priori for search. In this paper, we tackle these challenges in activity video retrieval through a combination of Relevance Feedback and Transfer Learning.

[1] This work was partially supported by NSF IIS 0712253 and the DARPA VIRAT program.

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 157–170, 2011.

Fig. 1. Example of similarity between two different classes. If training data for "volleyball" were abundant while training data for "basketball" were scarce, the knowledge on classifying volleyball could be used to supplement the basketball training process.

To deal with difficulties in training systems for the vast array of queries users could make, Relevance Feedback (RF) [15] can be used and has been effectively applied to image retrieval [24]. The idea is to first search a database with respect to an initial query and return retrieval results to the user. If the user is dissatisfied with the results, user feedback on the relevance of retrieved items may be provided. The system could then use the feedback to better learn what the user has in mind and return refined results. If the user is still dissatisfied, then another iteration of user feedback may be repeated and retrieval results refined until the user is satisfied. Since user feedback is provided in RF, it is possible build custom classifiers in an online fashion for the user. Thus a wide range of queries can be made without the need to train them a priori.

However, a drawback of RF is when used to search videos of complex activities, a large amount of user feedback may be needed for good performance. (In other words, the few rounds of feedback a user would tolerate would provide too scarce a training set.) Transfer Learning (TL) [14] is a Machine Learning formulation where knowledge learned from one or more classification tasks is transfered over to a target task where the target task training data is scarce. If the abundant training data of source task(s) are *related* to the target task, it can be used to bias the classifier for the target task so that generalization performance can be improved.

As an example, consider the related activities "volleyball" and "basketball" (see Fig. 1). Say we are interested in classifying whether videos are of "basketball" but the amount of training data available is very limited. If the amount of training data for the task of classifying "volleyball" or "not volleyball" were abundant, the knowledge from the "volleyball" classification task could be used to supplement the training of the "basketball" task in order to improve generalized accuracy on classifying "basketball" videos.

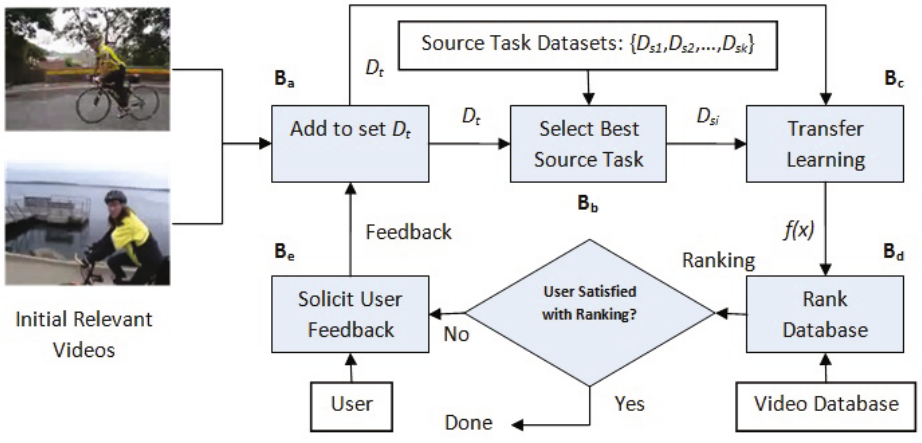


Fig. 2. Flowchart of system. Set D_t is initially empty before execution. After the first execution of block $\mathbf{B_a}$, set D_t should consist only of the initial relevant videos.

Provided system designers built a set of source task datasets for a variety of activities (this set of activities would only account for a small fraction of possible queries users could make), we could use the source data within a TL framework and combine it with RF to reduce the amount of needed user feedback. One of the key issues in combining RF and TL is determining which source task(s) are *related* to the target query, which is one of the main contributions of this work.

1.1 Overview and Contributions of Proposed Approach

Overview. We now provide and overview of our proposed approach. In our formulation, the user first submits a few example videos representing their target query that can be used as initial queries to start the RF process. This is a reasonable assumption as it should be possible for users to obtain some sample videos at least *similar* to what they have in mind. For example, if a user wanted to find videos of cross country cycling, a few example videos of people riding bicycles in general should suffice. Such initial seed results might be obtained through a text query which often generates only a few relevant examples, especially when videos are only sparsely tagged. For example, a text search on Google.com for "rally racing video game" videos results in some relevant footage being retrieved but the search results are also swamped with footage from "X Games" rally races (real-life sporting events). If the user cannot refine his text query to improve search results, he can select the few relevant examples and use them to start a RF loop to refine his search results.

The basic flow of our proposed system is as follows: (The following steps are annotated with corresponding blocks in Fig. 2.)

1. Let D_t be an empty set.
2. User submits a few initial query examples of relevant videos.
3. The initial query examples are added to set D_t. (Block $\mathbf{B_a}$.)
4. The source task datasets and D_t are then processed in our algorithm (Sec. 3.3) for finding the best source task to transfer from. (Block $\mathbf{B_b}$.)
5. The best source task's training dataset and training data in D_t are then used in TL to obtain a classifier f for ranking the video database. (Block $\mathbf{B_c}$.)
6. The classifier f is used to rank the video database. (Block $\mathbf{B_d}$.)
7. The top N ranked videos from the database are then shown to the user.
8. If the user is satisfied with the results, the process terminates. Otherwise the system solicits user feedback (details to follow later). (Block $\mathbf{B_e}$.)
9. The user feedback is added to set D_t (block $\mathbf{B_a}$) and the process continues from step 4.

The feedback strategy in step 8 is a simple but effective approach to RF based on Active Learning [18]. Rather than solicit feedback on retrieved items, $\text{SVM}_{\text{Active}}$ [20] showed effective performance in image retrieval when soliciting feedback on the items considered most ambiguously relevant by the system.

Contribution. Despite the effectiveness of $\text{SVM}_{\text{Active}}$ in image retrieval, the complexity of video activities limits the effectiveness of this framework. The main contribution of our work is in extending $\text{SVM}_{\text{Active}}$ to use TL for incorporating prior knowledge thus decreasing the required amount of user feedback. One of the key issues in combining RF with TL is in deciding what source task to transfer to the target task and we offer a solution in Sec. 3.3. As we explain in Sec. 2, our work is also one of the first to explore combining RF and TL. As we show in experiments on the YouTube Action Dataset [10], our framework provides benefits in improved ranking performance over standard RF frameworks for retrieval of complex activities.

2 Relation to Existing Work

Existing work in activity recognition demonstrates a trend of moving toward more complex activities. [8, 9, 10, 17] The main approaches of such work is to use new features, feature pruning techniques, and classification methods for improved complex activity recognition. However when applied to video retrieval, the subjectivity of human users is not modeled in these approaches. As a result, we propose a RF method that addresses this issue.

We note the main goal of this work is not to improve over previous work in terms of raw accuracy in activity recognition. Our focus is on the mechanism for quickly learning a user's subjective notions of activity class membership through user interaction. In fact, current work on designing features and algorithms for activity recognition is complementary to our work and could be integrated into our framework for overall improved retrieval. We now provide a review of related work in the two core tasks of our RF and TL framework for activity retrieval.

2.1 Relevance Feedback for Video Retrieval

In early work [11], RF for video was implemented by allowing the user to set weights in a scoring scheme utilizing various video features. In [4], this idea was extended to adaptively tune weights in a color and motion scoring scheme based on RF on top ranked videos. Other work [7] utilized more features such as speech recognition text, color, and motion in a weighted scoring scheme where weights were adaptively tuned based on RF of retrieved videos. In addition, semantic concept (e.g. "car") weightings were learned. A departure from the use weighted scores can be found in [13] where different scoring algorithms were adaptively chosen at each iteration of user feedback. However, the adaptive selection of scoring algorithms had to be manually trained by expert human users.

These systems showed good performance in their results. However, some can be complex, employing many different components. In applying them on larger scale problems, tuning the many parameters involved could be a daunting task.

Furthermore, most of the approaches described in this section do not make use of prior knowledge from the world to decrease the required amount of user feedback. While [7] used prior knowledge by explicitly building in high-level concepts like "cars", this approach requires learning a large number of classes that still would not cover the full range of queries users could make. We therefore address this issue by integrating TL into RF so that auxiliary training data of *different* classification problems from the target query can still be used to introduce prior knowledge into the system's learning process.

2.2 Basics of Transfer Learning

Before discussing related work in TL, we introduce a few TL concepts to provide context. In TL, there can be different relationships between the source and target tasks. Let task S be the source task and D_s be the source training set and task T be the target task and D_t be the target task's training set (where $|D_s| >> |D_t|$). Then TL can be subclassed into the following scenarios of interest:

1. S and T classify for the same class (e.g. running) but the distributions over the data for S and T are not the same. This is called the *Cross-Domain* problem in some work. As an example, if the training data D_s had been collected with camera A and D_t had been collected with camera B, simply combining D_s with D_t to improve classification accuracy on videos taken with camera B may not work well. (The cameras may have been positioned differently or have other differing characteristics.) The goal is to adapt the knowledge from D_s to augment the knowledge from D_t.
2. S and T classify for different but related classes. For example, S could be "volleyball" and T could be "basketball" (see Fig. 1). Since task S is related to task T, it should be possible to use the knowledge learned from D_s to improve generalization on D_t. This is the problem we focus on in this work.

There are more relationships between source and target tasks in TL described in [14] but the above mentioned ones are the most pertinent to our discussion.

2.3 Transfer Learning with Multiple Source Tasks

TL has been shown to be effective in transferring knowledge when source and target tasks are related. However, when there are multiple source tasks, deciding which to transfer from is still a difficult problem [14]. If a source task is too unrelated to the target task, transferring from such a source may result in *negative transfer* (transferring knowledge hurts target classification performance). The following work addresses TL in the presence of multiple source tasks.

In [23], the authors offer two methods for learning from multiple source datasets where some source tasks can be unrelated to the target. One method is effective but inefficient. The other finds a weighted linear combination of source classifiers and is efficient but only shows benefits when target data is very scarce.

In [21,22], they propose the Adaptive-SVM (A-SVM) for regularizing a target Support Vector Machine (SVM) [1] hyperplane to be similar to a related source hyperplane while still fitting the scarce target training data. The problem they focus on is the *Cross-Domain* problem (see Sec. 2.2). For example, the detection of concepts such as "weather" between news programs on different TV stations. The editing style and camera work of different TV stations causes the data for the same classes to be distributed differently. In addition to transferring knowledge from related tasks, they also explore determining which source tasks would result in *positive transfer*. To achieve this, they determine which source classifiers have the best estimated performance on the target class. Since we use the $\text{SVM}_{\text{Active}}$ approach [20], the TL described in this work is most related to our focus. Thus we extend the ideas from [21,22] beyond the Cross-Domain case.

Recent work related to A-SVMs [3, 6], present new mechanisms for Cross-Domain transfer of video actions and events. However, they do not present methods for source task selection. Furthermore, these mechanisms were designed for Cross-Domain transfer which may not be directly applicable to our problem of general TL. As the focus of the TL component in our work is in source task selection, we leave investigations into the possibility of adapting the transfer mechanisms in [3, 6] to general TL for future work. Finally, the related work mentioned here do not interact with the user which as mentioned before is crucial for capturing user subjective views of relevance.

2.4 Transfer Learning for Relevance Feedback Search

To the best of our knowledge there is no work on the general use of TL in RF. The RF surveys [5, 16, 24] do not even mention TL being applied to RF. Recent related work in the literature is mainly concerned with the Cross-Domain transfer problem for RF.

In [19], a study on how social tagged images could aid video search is presented. Their work is mainly concerned with how well manual relabeling of social tagged images without adaptation would work in a Cross-Domain scenario for video retrieval. They show results using RF and the benefits of simply cleaning up noisy labels without using adaptation. This framework does not apply in our case since we are working in a more general TL scenario.

In [12], two Cross-Domain learning methods are presented for RF. The first method uses a linear combination of the source and target classifier outputs with equal weighting. The second involves solving a regularized regression problem. Both methods performed similarly but combining the two via a heuristic for which method to use for each iteration of RF gave better overall performance.

While there is a little work on combining Cross-Domain transfer and RF in the literature, Cross-Domain transfer is only a special case of TL. The type of TL we explore involves transfer from different but *related* classification tasks and we offer a means of automatically determining task relatedness. Thus we present a complete system for RF search based on general TL. As stated earlier, this will also be one of the first explorations in combining RF and TL.

3 Relevance Feedback Using Transfer Learning for Activity Search

3.1 Scoring Videos and Relevance Feedback

In this work, we assume that videos can be represented as fixed length vectors of extracted feature histograms such as STIP [9]. These vectors could then be used in SVM training of classification tasks. Once trained, the relevance *score* of a video is interpreted as its distance to the SVM decision surface where the higher the score, the more relevant a video. For example, if we used a linear SVM for scoring, we would have $score(\mathbf{x_i}) = \mathbf{w} \cdot \mathbf{x_i} + b$ where $\mathbf{w}$ is the normal to the SVM hyperplane, $\mathbf{x_i}$ is a video from the database, and b is the bias term.

Following the $\text{SVM}_{\text{Active}}$ framework [20], our system solicits feedback on the N videos the system finds most ambiguously relevant (those nearest the SVM hyperplane) and the user labels these videos as either relevant or irrelevant. Once relevance labels have been solicited from the user, the system can use the additional labels to retrain a more accurate classifier. This classifier could then be used to assign a new score to each video in the database and rerank them to better fit the user's target query. Our work extends $\text{SVM}_{\text{Active}}$ by incorporating TL. We now describe the components of our TL system.

3.2 Transferring Knowledge from a Source Task

Let D_s and D_t be the training data for source task S and target task T respectively. (Where $|D_s| >> |D_t|$.) Then ideally if the source and target tasks were the same, we could just train a more powerful classifier for the target task by augmenting D_t with D_s. In practice, the source and target tasks are unlikely to be the same but they could still be related. Then we could still augment D_t with D_s but with less weight given to the data in D_s.

We accomplish this by adjusting the C parameter in the SVM formulation. Recall that training an SVM involves solving the following optimization problem:

$$\min_{\mathbf{w},\xi} \{\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i\} \tag{1}$$

$$\text{s.t. } y_i(\mathbf{x_i} \cdot \mathbf{w} + b) - 1 + \xi_i \geq 0, \xi_i \geq 0$$

where $\mathbf{x_i}$ is the i^{th} datapoint and y_i, ξ_i are the label and slack variable associated with $\mathbf{x_i}$. $\mathbf{w}$ is the normal to the hyperplane. C is the parameter that trades off between training accuracy (high C) and margin size (low C).

Let D_{aug} be D_t augmented with D_s and let the data from D_s be indexed from 1 to n in D_{aug} while the data from D_t be indexed from $n+1$ to $n+m$ in D_{aug}. Then to weight the source data and target data in the SVM training of D_{aug} we solve the following:

$$\min_{\mathbf{w},\xi} \{\frac{1}{2}\|\mathbf{w}\|^2 + C_s \sum_{i=1}^{n} \xi_i + C_t \sum_{i=n+1}^{n+m} \xi_i\} \tag{2}$$

$$\text{s.t. } y_i(\mathbf{x_i} \cdot \mathbf{w} + b) - 1 + \xi_i \geq 0, \xi_i \geq 0$$

where all the variables are as described in Eq. 1 and C_s and C_t are the different parameters trading off the "hardness" versus "softness" of fitting the associated datapoint. (Note that we set $C_s < C_t$.)

We note that there was little difference between using all the source data to bias the target SVM and using just the support vectors from the source SVM. Since using only the support vectors results in faster training speeds, we train only on the source task support vectors in our implementation.

The A-SVM [21,22] could have been used in place of this section's proposed method of transfer (which they call the "aggregate approach"). However the A-SVM does not offer benefits in improved accuracy over the aggregate approach and can even perform worse in some tests. The main advantage of using A-SVM is shortened training time. As the focus of this paper is on the feasibility of combining RF and TL for improved accuracy and the aggregate approach is more standard, we chose to use the aggregate approach.

3.3 Determining Which Source Task to Transfer From

Sec. 3.2 assumed we knew which source classifier to transfer from. However, transferring from the wrong classifier can hurt performance on the target task.

In [21], a number of strategies for choosing which source classifier to transfer from were presented. One method was to use score aggregation from multiple source classifiers. The basic idea was to use the "average" of multiple source classifiers with the hope that this would result in a more accurate classifier for assigning pseudo-labels to the unlabeled data. These pseudo-labels would then be used to evaluate how much individual source classifiers help improve ranking performance on the unlabeled examples. This approach does not work in our case. Since the authors were transferring knowledge in a Cross-Domain setting, all the source classifiers were assumed to classify for the same class. In our case, the source classifiers can be very unrelated to each other and thus combining an "average" of the source classifiers results in very poor performance.

Another proposed method was to assign scores to all unlabeled items using a potential source classifier (one trained on source data) and use the Expectation

Maximization (EM) algorithm to fit two Gaussian components to the scores. If the scores separate the data well then the means of the found Gaussian components should have greater distance between them. While a good idea, this is still not directly applicable to our problem because the target data are never used in this process; thus the same source classifier would always be selected regardless of the user feedback. However, if we first transfer the source classifier to the target classifier and then use the resulting classifier to score the unlabeled data, EM can be used to determine how well the transferred classification separates the data. We use this new procedure for determining which source classifier would help the target classifier produce the best separation of items in the database.

Formally, let D_s and D_t be the source and target training data and let $TL(D_s, D_t)$ be a function that produces a classifier where D_s was used to transfer knowledge to the target task (as described in Eq. 2). Then the following steps are taken to evaluate the quality of using D_s for the transfer:

1. Produce SVM $T_s = TL(D_s, D_t)$.
2. Use SVM T_s to compute scores (Sec. 3.1) Sc on the unlabeled database.
3. Use EM to fit Gaussian components $\mathcal{N}(\mu_1, \sigma_1^2)$ and $\mathcal{N}(\mu_2, \sigma_2^2)$ to scores Sc.
4. Determine the distance $d_\mu = (\mu_1 - \mu_2)^2$.

The distance d_μ can be used to indicate how well transferring the given source task to the target task would separate the unlabeled data (larger values are better). This provides an indication of whether the source task helps improve target task classification. The same procedure can be used to score the transfer for each of the available source tasks and the best source task could be chosen as the one to transfer from. We call this the **Score Clustering (SC)** method.

We note that projecting all source training data onto the subspace of the unlabeled database was found to be a helpful preprocessing step for determining what to transfer. Thus we first performed Principal Components Analysis on the unlabeled videos to obtain a set of basis vectors $\mathbf{V}$. We then projected *all* source task videos *and* unlabeled videos onto $\mathbf{V}$. So in our implementation, the projected videos were used instead of the original STIP histograms in all learning components of our system.

3.4 Integrating Relevance Feedback with Transfer Learning

We now formally describe the process of selecting a source task and transferring knowledge to the target task (user query) in the RF framework. Let $S_{Dtasks} = \{D_{s1}, D_{s2}, ..., D_{sk}\}$ be the set of source task training sets and D_t be the target task's training set. The TL portion of our framework operates as follows:

1. Given training data D_t from user feedback, determine the best source task training data D_{si} from set S_{Dtasks} to transfer from using SC (Sec. 3.3).
2. Use D_{si} to bias the learning of D_t using Eq. 2 and produce an SVM T_{si}.

SVM T_{si} is then used to rank the database of videos and if needed, feedback will be solicited on videos nearest the hyperplane of T_{si}. (Note that on each iteration of feedback, the choice of which task to transfer from is revisited.)

4 Experiments

Feature Representation and SVM Training. We first converted all videos into fixed length vectors representing histograms of STIP features [9]. The first step to getting these histograms was to build a codebook of STIP features. We did so by taking 100,000 random STIP features from videos and using K-means to identify 1,000 centers. The set of centers were then treated as the codebook. Afterward, for each video in our experiments, we extracted its STIP features, quantized them according to the codebook, and created a 1,000 dimensional vector with counts how many occurrences of each type of quantized STIP feature was present in the video. For SVM training, we used the SVM and Kernel Methods Matlab Toolbox [2] and selected the linear kernel as it provided sufficient accuracy for our study.

Dataset. We used the YouTube Action Dataset [10] in our experiments. This dataset consists of about 1,600 videos collected from YouTube.com with 11 categories of actions ranging from "basketball shooting" to "dog walking." Its videos are very challenging as they were taken outside of controlled settings and feature camera shake, differences in lighting, video quality, and camera position.

We note that in [10], their goal was to obtain *high classification accuracies* of video activities through new feature extraction and pruning techniques. Here, we are *not* attempting to obtain the best performance in terms of classification accuracies. Instead we are aiming to obtain the *best improvement* in performance through the use of TL. More sophisticated feature extraction and classification algorithms could be used in our framework but we chose to use standard features and learning algorithms so as to establish a control in our experiments.

Experimental Setup. We chose all videos in the classes basketball, biking, diving, golf swing, and horse riding to be in our unlabeled database and all remaining videos to be source data. For TL, we set $C_s = 10^{-4}$ and $C_t = \infty$ in Eq. 2. There were a total of 778 videos in our unlabeled database with on average 150 videos per class. The source data was used to define a set of 1-versus-all classification problems (for example volleyball versus not volleyball). The target queries were for distinguishing one of the five classes listed above from the total unlabeled database. Feedback was seeded with five randomly selected positive and five randomly selected negative examples. Each query session involved three rounds of simulated user feedback where 10 examples nearest the SVM hyperplane would be labeled. By simulated, we mean that ground truth labels were used to judge the relevance of videos. In future work, we plan to compare system performance with simulated and real user feedback. Note that iteration 1 in the results only uses the initial examples from the user. Iteration 2 is when feedback is first used. Thus by iteration 4, the user would only have given feedback on $\mathbf{30/778} \approx \mathbf{4}\%$ of the database.

We also ran experiments on a variant of our system where no TL was used. That is, we replaced blocks $\mathbf{B_b}$ and $\mathbf{B_c}$ in Fig. 2 with a single block that only takes in the target training data D_t and trains an SVM for it. In addition to testing against the no TL case, we also tested against a straightforward heuristic

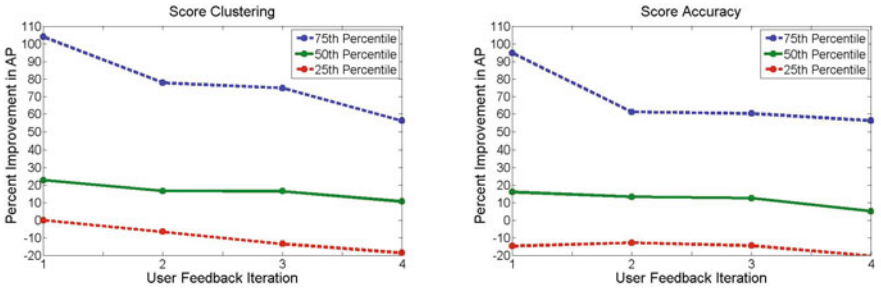


Fig. 3. Plots of percent improvement in AP of TL over not using TL for two methods of choosing source task: Score Clustering (left) and Score Accuracy (right). The distribution of improvements over all tasks and all initial video inputs are shown. Quartiles are plotted since percent improvements are highly varied but skewed toward positive values. The 50th percentile indicates the median percent improvement for a given iteration. The left graph's 75th percentile mark in iteration 1 indicates that 25% of the test queries had percent improvements over 100%. Note that iteration 1 only uses initial seeded examples from the user. Iteration 2 is where user feedback is first incorporated.

for source task selection. (To compare against our SC method.) If a source task S and target task T are related, we would expect TL from S to T to improve performance. Thus we did a set of experiments where $\mathbf{B_b}$ from Fig. 2 was replaced with the following procedure:

1. Given target task training data D_t, train an SVM T_t.
2. Determine the classification error of each source task training set D_{sj} with respect to SVM T_t.
3. Choose training set D_{si} with the lowest error as the source to transfer.

The intuition is if tasks S and T are related, using a classifier trained on one task's training data to classify the other should result in less degradation than if the tasks were not related. We call this the **Score Accuracy (SA)** method.

Metrics Used. As different combinations of initial examples can affect performance, we tested querying for each category 100 times. (With the same initial queries used for both TL and non-TL tests.) We computed Average Precision (AP) to assess ranking performances (on only the currently unlabeled videos) for each iteration of feedback as:

$$AveragePrecision = \frac{1}{num}\sum_{r=1}^{N}(P(r) \times \mathrm{rel}(r)) \tag{3}$$

where $N = 50$ in our experiments, $P(r)$ is the precision at rank r, and $rel(r)$ is the indicator function for whether the r^{th} item in the ranking is relevant. We set $num = 50$ so AP values range from 0.0 to 1.0 with 1.0 being an ideal ranking.

A natural way to measure overall improvement from TL for all target queries would be to determine the average percent improvement in AP between corresponding TL versus no TL tests. However we found that although a majority of

Table 1. MAP for Different Queries (row) over Feedback Iterations (col.) The source tasks were soccer juggling, swing, tennis swing, trampoline jumping, volleyball spiking, and dog walking

Transfer Learning				
Feedback Iteration	1	2	3	4
Basketball	0.26 ± 0.12	0.31 ± 0.14	0.34 ± 0.13	0.35 ± 0.12
Biking	0.55 ± 0.15	0.63 ± 0.14	0.70 ± 0.13	0.74 ± 0.13
Diving	0.21 ± 0.16	0.25 ± 0.15	0.29 ± 0.15	0.31 ± 0.15
Golf Swing	0.21 ± 0.12	0.26 ± 0.15	0.29 ± 0.17	0.26 ± 0.18
Horse Riding	0.19 ± 0.07	0.30 ± 0.11	0.40 ± 0.13	0.46 ± 0.13
No Transfer Learning				
Feedback Iteration	1	2	3	4
Basketball	0.17 ± 0.11	0.25 ± 0.13	0.28 ± 0.13	0.29 ± 0.13
Biking	0.49 ± 0.11	0.59 ± 0.12	0.66 ± 0.12	0.72 ± 0.11
Diving	0.13 ± 0.13	0.18 ± 0.15	0.24 ± 0.15	0.28 ± 0.15
Golf Swing	0.14 ± 0.12	0.16 ± 0.14	0.19 ± 0.15	0.23 ± 0.16
Horse Riding	0.21 ± 0.09	0.28 ± 0.12	0.34 ± 0.15	0.41 ± 0.16

our tests resulted in positive transfer, there was a large amount of variation in percent improvement. For example, in one case we observed a AP value of 0.0016 for no TL but with TL, we obtained a AP of 0.4. In other cases, we observed improvements in AP of +0.2. So determining means and standard deviations in percent improvement does not adequately summarize our results.

Thus we plotted quartiles over *all* observed percent differences in our tests across the feedback iterations (Fig. 3) as this more adequately illustrates how our percent improvements in AP were distributed. The 50th percentile marks on the figure are the median percent improvements (as a function of feedback iterations) observed from all of the test runs conducted. The median line in the score clustering (SC) method's results indicates that half of all tests conducted resulted in at least about 20% improvement. The 25th percentile mark in the first iteration of the SC graph indicates that 75% of the tests resulted in some improvement from TL. Similarly, the first iteration 75th percentile mark in the SC graph shows that 25% of tests run resulted in over 100% improvement.

Results. Fig. 3 indicates that SC is better than SA (see Sec. 4) in determining which source task to transfer from. This is probably because the SC method attempts to find which source task's bias would improve classification with respect to the target data on the particular unlabeled database being searched. So SC does not attempt to transfer knowledge for generalized performance and instead bases its criterion on the data being searched instead. The SA method does not consider any of the unlabeled data in the database which limits its ability to find a source task good for separating data on the database of interest. It is also not surprising that percent improvement tends to drop as the amount of user

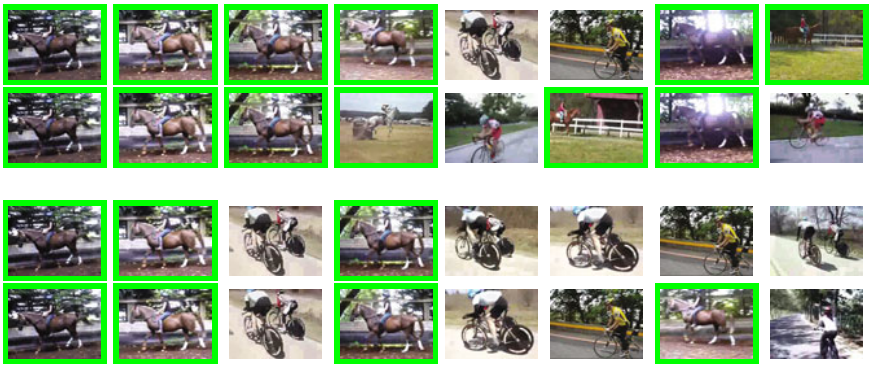


Fig. 4. Ranking results for "Horse Riding." The first two rows show the top 8 videos for the first and second feedback iterations with TL. The bottom two rows are the first and second feedback iterations without TL. With TL, there is less confusion between biking and horse riding *and* a greater variety of relevant videos are captured.

feedback is increased. As the amount of target task training data increases, one would expect the target classifier to generalize better without the need for TL.

While we could not show meaningful averages and standard deviations for individual percent improvements, we can show the overall Mean AP (MAP) for each class query to give readers a concrete idea of how MAP improves over feedback iterations. Results for TL (using SC for source task selection) and no TL are shown in Table 1. Fig. 4 also shows sample results for retrieval of "horse riding" videos for the first two user feedback iterations of the TL and no TL cases. (More such results are provided in the supplementary materials.)

5 Conclusion

We presented a framework in RF for complex activity video retrieval through a combination of RF and TL and demonstrated its utility on a real-life dataset of Internet videos. The primary contribution of this work was the use of EM to determine the best source task data to use for knowledge transfer resulting in overall less required user feedback in the search process. We also made one of the first explorations of combining RF with general TL. As the key problem in this framework is the choice of source task data to transfer, we hope to improve on our current results in the future through improvements in source task selection.

References

1. Burges, C.: A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2, 121–167 (1998)
2. Canu, S., Grandvalet, Y., Guigue, V., Rakotomamonjy, A.: SVM and kernel methods matlab toolbox. Perception Systmes et Information, INSA de Rouen, Rouen, France (2005)

3. Cao, L., Liu, Z., Huang, T.: Cross dataset action detection. In: CVPR. IEEE, Los Alamitos (2010)
4. Chen, L., Chin, K., Liao, H.: An integrated approach to video retrieval. In: ADC, Australian Computer Society, Inc. (2008)
5. Crucianu, M., Ferecatu, M., Boujemaa, N.: Relevance feedback for image retrieval: a short survey. State of the art in audiovisual content-based retrieval, information universal access and interaction including data models and languages, DELOS2 Report (FP6 NoE) (2004)
6. Duan, L., Xu, D., Tsang, I., Luo, J.: Visual event recognition in videos by learning from web data. In: CVPR. IEEE, Los Alamitos (2010)
7. Hauptmann, A., Lin, W., Yan, R., Yang, J., Chen, M.: Extreme video retrieval: joint maximization of human and computer performance. In: MULTIMEDIA. ACM, New York (2006)
8. Hu, Y., Cao, L., Lv, F., Yan, S., Gong, Y., Huang, T.: Action detection in complex scenes with spatial and temporal ambiguities. In: ICCV. IEEE, Los Alamitos (2009)
9. Laptev, I.: On space-time interest points. International Journal of Computer Vision 64, 107–123 (2005)
10. Liu, J., Luo, J., Shah, M.: Recognizing realistic actions from videos "in the wild". In: CVPR. IEEE, Los Alamitos (2009)
11. Liu, X., Zhuang, Y., Pan, Y.: A new approach to retrieve video by example video clip. In: MULTIMEDIA. ACM, New York (1999)
12. Liu, Y., Xu, D., Tsang, I., Luo, J.: Using large-scale web data to facilitate textual query based retrieval of consumer photos. In: MULTIMEDIA. ACM, New York (2009)
13. Luan, H., Zheng, Y., Neo, S., Zhang, Y., Lin, S., Chua, T.: Adaptive multiple feedback strategies for interactive video search. In: CIVR. ACM, New York (2008)
14. Pan, S., Yang, Q.: A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering (2009)
15. Rocchio, J.: Relevance Feedback in Information Retrieval, pp. 313–323. Prentice-Hall, Inc., Englewood Cliffs (1971)
16. Ruthven, I., Lalmas, M.: A survey on the use of relevance feedback for information access systems. Knowledge and Engineering Review 18, 95–145 (2003)
17. Ryoo, M., Aggarwal, J.: Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In: ICCV. IEEE, Los Alamitos (2009)
18. Settles, B.: Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison (2010)
19. Setz, A., Snoek, C.: Can social tagged images aid concept-based video search? In: ICME. IEEE, Los Alamitos (2009)
20. Tong, S., Chang, E.: Support vector machine active learning for image retrieval. In: MULTIMEDIA. ACM, New York (2001)
21. Yang, J., Yan, R., Hauptmann, A.: Cross-domain video concept detection using adaptive SVMs. In: MULTIMEDIA. ACM, New York (2007)
22. Yang, J., Hauptmann, A.: A framework for classifier adaptation and its applications in concept detection. In: MIR. ACM, New York (2008)
23. Yao, Y., Doretto, G.: Boosting for transfer learning with multiple sources. In: CVPR. IEEE, Los Alamitos (2010)
24. Zhou, X., Huang, T.: Relevance feedback in image retrieval: A comprehensive review. Multimedia Systems 8, 536–544 (2003)

A Compositional Exemplar-Based Model for Hair Segmentation

Nan Wang[1], Haizhou Ai[1], and Shihong Lao[2]

[1] Computer Science & Technology Department, Tsinghua University, Beijing, China
ahz@mail.tsinghua.edu.cn
[2] Core Technology Center, Omron Corporation, Kyoto, Japan
lao@ari.ncl.omron.co.jp

Abstract. Hair is a very important part of human appearance. Robust and accurate hair segmentation is difficult because of challenging variation of hair color and shape. In this paper, we propose a novel Compositional Exemplar-based Model (CEM) for hair style segmentation. CEM generates an adaptive hair style (a probabilistic mask) for the input image automatically in the manner of Divide-and-Conquer, which can be divided into decomposition stage and composition stage naturally. For the decomposition stage, we learn a strong ranker based on a group of weak similarity functions emphasizing the *Semantic Layout similarity* (SLS) effectively; in the composition stage, we introduce the *Neighbor Label Consistency* (NLC) Constraint to reduce the ambiguity between data representation and semantic meaning and then recompose the hair style using alpha-expansion algorithm. Final segmentation result is obtained by Dual-Level Conditional Random Fields. Experiment results on face images from Labeled Faces in the Wild data set show its effectiveness.

1 Introduction

In computer graphics, hair acquisition [1] [2] and hair geometry modeling [3] have achieved significant progresses. While in computer vision, hair style analysis or hair segmentation discussed in this paper is still an ongoing research issue. Hair is a very important part of human appearance especially in consumer images. In visual surveillance condition or criminal cases, face details usually cannot be seen or remembered or described clearly. However, hair style is easier to be identified and described in most cases, so it usually becomes one of the most important descriptors for some specific target person. For this application, hair segmentation becomes a necessary intermediate step to hair style identification. Moreover, with the rapid development of internet, online makeup has become more and more popular. When people want to see whether or not some hair style fits them, a good hair style identification or search tool could help a lot, which also makes hair segmentation necessary. Nevertheless, there are challenges for segmenting hair area in consumer images because of the variation of shape and color. Robust hair segmentation is by far an unsolved problem.

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 171–184, 2011.

Fig. 1. It is easy to tell bald from the long hair. But it is extremely hard to tell the long hair from longer ones.

Yacoob and Davis [4] build a hair color model and then adopt a region growing algorithm to modify the hair region. However, this method will only work when the hair color doesn't change significantly, especially for the dark hair. Consumer images do not fit in this constraint.

Lee et al [5] give a more practical algorithm for consumer images. They first cluster hair style and the color of hair and face into several typical patterns manually. And then for each hair style, choose the fittest hair and face color model and modify it according to the input image. A Markov Random Field is built and inferred to maximize the joint probability distribution of each pixel on each label. The one whose labeling result has the minimized distance to its corresponding hair style is chosen as the final hair style. Their work gives a practical idea to solve the problem; nevertheless there are still several issues we need to focus on. Hair style classification is a hard issue. It is difficult to decide how many patterns are appropriate even for just front view, let alone cases with side view. With a predefined cluster label, it is still hard to decide which hair style an input image belongs to. It may be easy to tell bald from long hair, but extremely hard to tell long hair from longer one, as shown in Figure 1. Unfortunately, this classification is vital because unary term plays dominant role in graph model [6].

In Borenstein and Ullman [7], a combined top-down and bottom-up algorithm is proposed to solve the problem of figure-ground segmentation. During top-down procedure, image fragments and the corresponding figure background labels are extracted from training data first and then used to optimally cover an object in a novel image to induce the final segmentation result. Wang and Tang [8] approached the problem of face photo-sketch synthesis and recognition. The input image is normalized and divided into overlapped rectangles. For each rectangle, K candidate patches from the training set are selected. A multi-scale Markov Random Field model is used for the selection of optimal combination of patches. Jolic et al. [9] model the spatial correlations in image class structure by introducing the Stel to make image models invariant to changes in local measurement, while sensitive to changes in image structure.

Inspired by these works, we build a Compositional Exemplar-based Model (CEM, section 2) for hair style generation, which could generate an appropriate hair style for the input image. In our paper, actually four labels are used:

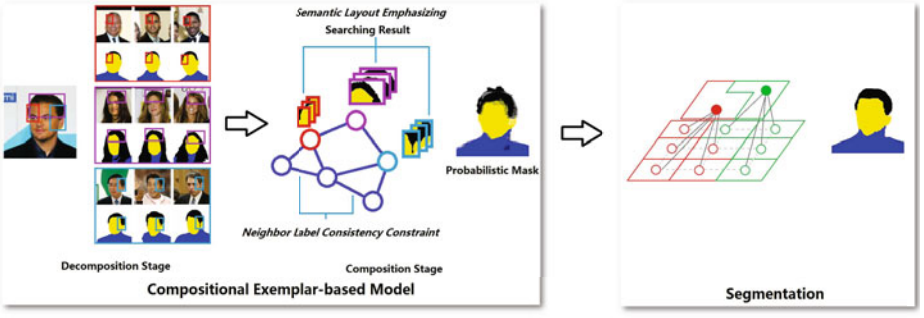


Fig. 2. Work flow of Compositional Exemplar-based Model. Color code for labels are: white - background, yellow - face, black - hair and blue - clothes. (This figure is best viewed in color.)

background, face, hair and clothes. CEM works in the Divide-and-Conquer way as illustrated in Figure 2.

In the decomposition stage, we design a group of *Semantic Layout Similarity* (SLS) features (section 2.1), which are combined together to get a strong and effective similarity function for each location respectively. Based on the similarity function, candidate segmentation results are collected for each local patch from a manually labeled library in this stage.

In the composition stage, we introduce a *Neighbor Label Consistency* (NLC) Constraint and organize local patches as a Markov Network (section 2.2). A well-defined consistency function promises the *regularity* [10], which allows us to optimize the CEM using α-expansion algorithm [10] [11] [12]. CEM finally generate a probabilistic mask as illustrated in Figure 2. With the favor of the mask, we obtain the final segmentation result using a dual-level Conditional Random Fields (section 3).

2 Compositional Exemplar-Based Model

It is hard to model the hair styles integrally, since hair styles have large variation as shown in Figure 1. The basic idea is to decompose a hair style into local patches and model each patch respectively. The reason is that although hair styles can differ from each other dramatically in global, they can still share some common *Semantic Layout* in local. In our paper, *Semantic Layout* means the actual label patterns of patches. There is intuitional evidence in the diagram of the Decomposition Stage in Figure 2. The purple patch of the input image covers forehead and hair root regions. The first three searching hair style are very different from the query one, but just in this local patch, they seem the same. This is why we can model hair style locally.

In the decomposition stage, candidate segmentation results are obtained independently. However, the independence of searching will lead to ambiguity sometimes, because we use the similarity defined in data representation level

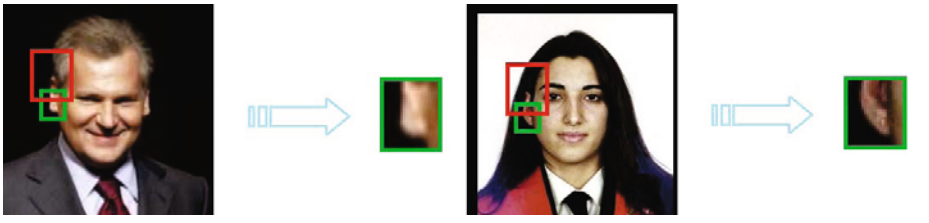


Fig. 3. Ambiguity of patches. Although the two green patches seem very similar to each other in date representation level, the dark color parts of the patches have totally different semantic meanings. When neighbor patch (in red) are considered together, the ambiguity can be avoided.

to approximate the actual one in semantic level. We give an example in Figure 3, where the local patches from two images are almost the same in the data representation level but have totally different meanings for the dark color part in the semantic level. However, if its neighbor patch (in the red rectangle) is considered together, this ambiguity can be avoided most of the time. From this point, we introduce the *Neighbor Label Consistency* (NLC) Constraint to reduce the ambiguity.

There are two key problems in the model. The first one is how to define a similarity to capture the *Semantic Layout* information. And the second one is how to select the best candidates for all patches together when NLC Constraint is introduced. They will be described in the next subsections respectively. Before that, we define some notations.

P_i is the local patch of the image and its corresponding label result is denoted as L_i. The local patches are required overlapping with its adjacent ones. Then the neighbor patch indices of P_i is denoted as $N(i)$. For each patch P_i, there is an exemplar library for it, which is denoted as $\left\{\mathcal{P}_i^k\right\}$. The manually labeled result for the exemplar library is $\left\{\mathcal{L}_i^k\right\}$. The *similarity function* in data representation level between patch P and Q is defined as $H(P,Q)$. The similarity function $C(P,Q)$ for *Semantic Layout* between patch P and Q is defined as follows:

$$C(P,Q) = \frac{1}{\Delta}\sum_m \delta\left(L_p^m = L_Q^m\right) \tag{1}$$

where Δ is the size of region P and Q. $\delta(\cdot)$ is Kronecker delta function.

2.1 Learning Similarity Function by SLS Features

In this subsection, we focus on how to define a similarity function to capture the *Semantic Layout* information. Similarity can be defined on the statistic information, such as histograms, or on the data structure, such as Euclid distance, or on a fusion of them. One thing should be noticed in the problem is that the feature compactness of different labels are not the same. For example, face and hair have some typical pattern of color or texture distributions; while clothes feature distribution is looser and background feature distributions barely share

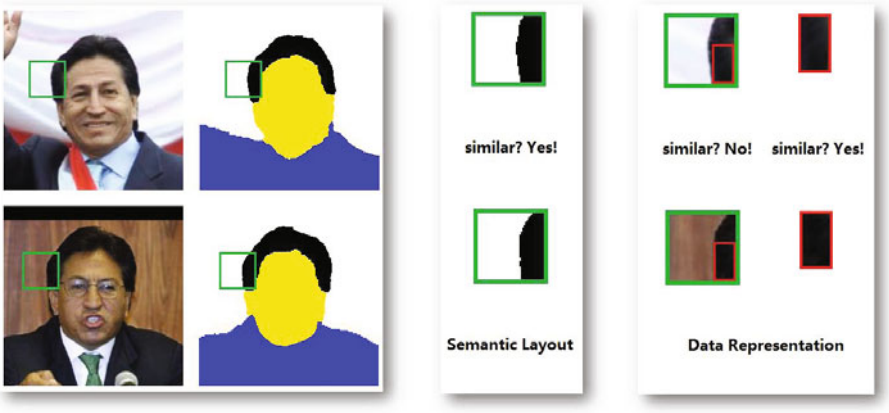


Fig. 4. Green patches of the two images have very similar *Semantic Layout*. However, if the similarity is defined based on the whole feature of the patch, background difference will dominate the similarity between them and cause a loss of the good exemplar candidate. Our SLS feature is calculated in selected sub-patches, such as the red rectangle. In this way, better consistency between data representation and semantic meanings can be achieved.

anything from one image to another. This characteristic will cause loss of good exemplar labels sometimes, as shown in Figure 4. To achieve a better consistency between data representation and semantic meanings, our similarity function is constructed based on the features in local sub-patches. A similar work to capture the *Semantic Layout* information is that of Shotton et al. [6] which presents a discriminative model to fuse shape, appearance and context information to recognize efficiently the object classes. Our algorithm is different from that since we focus on the explicit similarity of *Semantic Layout* of *patches*, while they focus on the classification of *pixel* using *Semantic Layout* as a learning cue.

Formally, denote the SLS feature set $\Phi = \{\phi_0, \phi_1, \cdots, \phi_M\}$. Our algorithm use color and texture as basic features, such as RGB, HSL color space, Gabor wavelet, which are represented as histograms (Gabor wavelet is transformed as LGBP [13](Local Gabor Binary Pattern)). Each SLS feature in Φ is determined by a triple $\langle F_m, R_m, B_m \rangle$. F_m denotes the feature type, which can be histogram of R channel in the RGB color space or LGBP in some specific frequency and orientation. R_m is the rectangle where F_m histogram is calculated. B_m is the bin index of the histogram. Let $\phi_m(\mathcal{P}_i) = \phi_{\{F_m, R_m, B_m\}}(\mathcal{P}_i)$ be the B_m bin value of F_m histogram extracted from $\mathcal{P}_i$ in the rectangle R_m. Then the weak ranker $h_m(\mathcal{P}_i, \mathcal{P}_j)$ is calculated as:

$$h_m(\mathcal{P}_i, \mathcal{P}_j) = -\frac{(\phi_m(\mathcal{P}_i) - \phi_m(\mathcal{P}_j))^2}{\phi_m(\mathcal{P}_i) + \phi_m(\mathcal{P}_j)} \tag{2}$$

which actually is the opposite number of one addend term from the Chi-Square Distance equation. So the final *similarity function* is a generalization of Chi-Square Distance.

Table 1. Preference Pairs Generation Algorithm

Input: Exemplar library in specific location $\left\{\mathcal{P}_i^k\right\}$, threshold τ
Output: Preference Pair Set (Training Set) $\{T_j\}$

- Initialization: $\{T_j\} \leftarrow \Phi$
- For each $\mathcal{P}_i^j$
  - Sort the other patches based on $C\left(\mathcal{P}_i^j, \mathcal{P}_i^k\right)$, and get a permutation of the other patches $\pi(m)$, which maps the patch's sorting index m to its original index $\pi(m)$ in $\left\{\mathcal{P}_i^k\right\}$.
  - For each $\mathcal{P}_i^{\pi(m)}$, which satisfies that $C\left(\mathcal{P}_i^j, \mathcal{P}_i^{\pi(m)}\right) < \tau$
    - Add $\left(\mathcal{P}_i^j, \mathcal{P}_i^{\pi(0)}, \mathcal{P}_i^{\pi(m)}\right)$ in $\{T_j\}$
  - End For
- End For

To get a good enough similarity function, we apply the RankBoost [14] learning algorithm to select the best SLS features and evaluate their weights. For the RankBoost algorithm, preference pairs should be defined to serve as the training data. In our problem, the preference is defined by the manually labeled result similarity of the two exemplar patches $C(\mathcal{P}_i, \mathcal{P}_j)$. For each exemplar library $\left\{\mathcal{P}_i^k\right\}$, one of them is used as the query patch, and the others are sorted based on $C(\mathcal{P}_i, \mathcal{P}_j)$. And we *prefer* that the similarity between the query one and the first one is larger than that between the query one and the end ones. Specifically, the preference pairs (training set) generation algorithm is shown in Table 1.

The training objective of our algorithm is to construct a strong ranker function (*similarity function* in our paper) so that:

$$\forall \left(\mathcal{P}_i^j, \mathcal{P}_i^{\pi(0)}, \mathcal{P}_i^{\pi(m)}\right) \in \{T_j\}, H\left(\mathcal{P}_i^j, \mathcal{P}_i^{\pi(0)}\right) > H\left(\mathcal{P}_i^j, \mathcal{P}_i^{\pi(m)}\right) \tag{3}$$

The *similarity function* $H(\cdot)$ is the weighted sum of weak rankers, the same as other boosting algorithm. The details of RankBoost training algorithm can be found in [14].

2.2 Introduce NLC Constraint into CEM

In CEM, NLC Constraint is achieved by enforcing pixel to be assigned the same label no matter which patch it locates in. So the *consistency function* can be defined by $C_A(P, Q)$, which is $C(P, Q)$ restricted on the overlapping area A of P and Q. The CEM can be represented formally as a Markov Network. The Node is the patch set $\{P_i\}$, and the neighborhood system is just defined before. Suppose C best candidate exemplars are reserved. The optimization of CEM can be done by minimizing the following energy function:

$$E(P) = \sum_i \left(\varphi_i(c_i) + \sum_{j \in N(i)} \varphi_{i,j}(c_i, c_j) \right) \tag{4}$$

where c_i denotes the index of exemplar that P_i finally take. The unary function $\varphi_i(c_i)$ is defined as:

$$\varphi_i(c_i) = -\log\left(H\left(P_i, \mathcal{P}_i^{c_i}\right)\right) \tag{5}$$

And the pairwise function $\varphi_{i,j}(c_i, c_j)$ is defined as:

$$\varphi_{i,j}(c_i, c_j) = -\log\left(C_A\left(\mathcal{P}_i^{c_i}, \mathcal{P}_j^{c_j}\right)\right) \tag{6}$$

However, the straightforward definition is not *regular* [10]. According to the theorem of [10], the *regularity* of pair wise term is a necessary and sufficient condition for graph-representability. So this energy function cannot be minimized by graph-cut based algorithm. The problem can be solved by expanding the node label set from $\mathcal{L} = \{0, 1, \cdots, C-1\}$ to $\mathcal{L} = \{0, 1, \cdots, nC-1\}$, where n is the vertices number. All possible candidate exemplars of all patches are grouped together. Since each patch i can only take label ranged between iC and $(i+1)C-1$ actually, the other assignment should be set as a maximum value. The mapping function between label index is $f(c_i) = c_i - iL$. The unary term is computed as:

$$\tilde{\varphi}_i(c_i) = \begin{cases} \varphi_i(f(c_i)) & iC \le c_i < (i+1)C \\ max & others \end{cases} \tag{7}$$

To satisfy the regular condition in [10], the pair wise term is modified as follows:

$$\tilde{\varphi}_{i,j}(c_i, c_j) = \begin{cases} \beta\varphi_{i,j}(f(c_i), f(c_j)) & iC \le c_i < (i+1)C, jC \le c_j < (j+1)C \\ 0 & c_i = c_j \\ max & others \end{cases} \tag{8}$$

The proof of the *regularity* of $\tilde{\varphi}_{i,j}(c_i, c_j)$ is given in supplementary files. With the constraint of unary term, c_i and c_j will always satisfy the first condition in pair wise term, when the assignment is optimal. So the labeling result of graph model with the expanding label set is equivalent to the former one. Although the expanding label set will increase computation load, in practice the inference is still fast enough, because the number of super pixels is very small in general. Denote the optimal solution of CEM as $\{L_i\}$. The probabilistic hair style mask is constructed to retain all the information of overlapping patches. The mask M is calculated as:

$$M_{i,l} = \frac{\sum_{i,j \in P_i} \delta\left(L_i^{\tilde{j}} = l\right) + \epsilon}{\sum_{i,j \in P_i} 1 + \epsilon} \tag{9}$$

where $M_{i,l}$ denotes the probability of assigning pixel i with label l. $\mathcal{L}_i^{\tilde{j}}$ is the manually labeled result of optimal exemplar in index $\tilde{j}$ which is the corresponding index of pixel j in patch P_i.

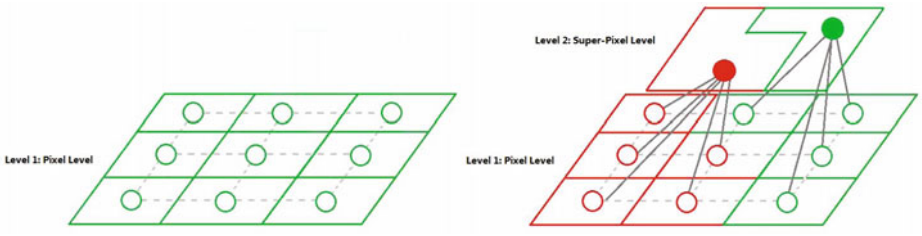


Fig. 5. Diagram of Dual-Level Conditional Random Fields

3 Segmentation with Dual-Level Conditional Random Field

Conditional Random Filed with higher order constrained has been used in segmentation problem and gets significant achievement recent years [15] [16] [17]. In this paper, we use a dual-level CRF to incorporate higher order constraint from super pixels obtained by JSEG [18].

There are two level vertices in the graph model of dual-level CRF. Vertices in level 1 are pixels in images and vertices in level 2 are super pixels produced by JSEG [18]. The structure of dual-level CRFs is illustrated in Figure 5. The edges only exist between vertices in level 1 and vertices between the two levels, while there are no edges between vertices in level 2, because superpixels are used as soft constraint in our model and final labeling results are obtained from level 1. The energy function is defined as follows:

$$E(x) = \sum_{i} \phi_i(x) + \sum_{i,j \in N(i)} \phi_{i,j}(x_i, x_j) + \sum_{i} \tilde{\phi}_i(x_i) + \sum_{i, i \in R_j} \tilde{\phi}_{i,j}(x_i, x_j) \quad (10)$$

where x_i is the label assigned to corresponding pixel or super pixel. ϕ_i and $\phi_{i,j}$ is the energy term defined on pixel level. $\tilde{\phi}_i$ is the super pixel unary term. $\tilde{\phi}_{i,j}$ is the pair wise term between pixel and its corresponding super pixel, which represent the higher order constraint by super pixel.

$\phi_i(x_i)$ and $\tilde{\phi}_i(x_i)$ have similar definition.

$$\phi_i(x_i) = \omega_{mask}\phi_i^{mask}(x_i) + \omega_{color}\phi_i^{color}(x_i) \quad (11)$$

where $\phi_i^{mask}(x_i) = -\log(M_{i,x_i})$. $\phi_i^{color}(x_i)$ is defined as the minus log of probability that current pixel's color in color distribution for the x_i label. In our experiment, the color distribution is represented as histograms. In $\tilde{\phi}_i(x_i)$, the mask probability is the average of probabilities of the pixels in its corresponding superpixel, and the color probability is defined as the similarity between color histogram of superpixel's and corresponding label's.

$$\phi_{i,j}(x_i, x_j) = \gamma \exp\left(-\beta \|I_i - I_j\|^2\right) \delta(x_i \neq x_j) \quad (12)$$

where β is set as $\left(2\left\langle \|I_i - I_j\|^2 \right\rangle\right)^{-1}$. γ is the model parameters.

Assuming i in level 1, j in level 2 and pixel i belongs to super pixel R_j , $\tilde{\phi}_{i,j}(x_i, x_j)$ is defined as Potts Model:

$$\tilde{\phi}_{i,j}(x_i, x_j) = \begin{cases} 0 & x_i = x_j \\ \tilde{\gamma} \exp\left(-\tilde{\beta} |R_j|\right) & other \end{cases} \tag{13}$$

where $|R_j|$ is the cardinality of super pixel R_j. $\tilde{\beta}$ is the inverse of the average over all super pixel sizes. $\tilde{\gamma}$ is model parameters just like γ. We use α-expansion algorithm to get the labeling result of the dual-level CRF.

4 Experimental Result

First of all, we label the training data, also called exemplar library, manually. Training data comes from Labeled Faces in the Wild database [19]. The reason for choosing this database is that images in LFW are general consumer images that are much less constraint than those used in face recognition researches, which is very good for validating the proposed algorithm. We manually labeled 1026 images. For each image, each pixel is assigned a label from the label set: background, hair, face or clothes. These images are divided into two halves randomly. One of them is used for learning similarity function and the parameters of CEM. The other is used for testing. The training and testing procedure is shown in Table 2 and Table 3 respectively. These dividing, training and testing procedure are repeated 10 times to get the experiment data.

The parameters of CEM are determined empirically. In consideration of speed, images are normalized as 72×72 and divided into 16×16 patches with step of 8 in both x and y directions. R_m in the training data are rectangles with sizes of 4×4, 8×8 and 16×16. The threshold is set as $\tau = 0.5$. The candidate number is set as $C = 10$. β in formula 8 is 8. For the CRF model parameters, $\gamma = \tilde{\gamma} = 8$, $\omega_{mask} = 1.6$ and $\omega_{color} = 0.4$. Both the max in formula 7 and 8 are set as 1000 to prevent an invalid inference result.

In Figure 6, we show some segmentation result by our algorithm. Hair style changes from bald to long and in different colors, it can be seen that our algorithm works robustly in the condition of large variation of hair shape and color and clutter background.

It takes us about 95 hours to train the SLS-based rankers. The training algorithm is applied independently for each local patch. So it can be extended on a distributed system easily to shorten the training time. To show the effectiveness of our similarity function, we used Normalized Discounted Cumulative Gain (NDCG) [22] to estimate the ranking quality. For a list of images sorted in descending order of the scores output by a learned ranking model, the NDCG score at the m-th image is computed as:

$$N_m = C_m \sum_{j=1}^{m} \frac{2^{r(j)} - 1}{\log(j+2)} \tag{14}$$

Table 2. Training algorithm

- Detect Face [20] and Eye locations [21] for each image and normalize it in the same size
- For each location, extracted the exemplar patch set $\{\mathcal{P}_i^k\}$
  - Generate training data as Table 1.
  - Generate strong ranker using RankBoost algorithm [14].
- End For

Table 3. Testing algorithm

- Detect Face [20] and Eye locations [21] for each image and **normalize it in the training size**
- For each location, extracted the exemplar patch set $\{\mathcal{P}_i^k\}$
  - Using strong ranker $H(\cdot)$ for current location to sort $\{\mathcal{P}_i^k\}$
  - Keep C exemplars as candidates
- End For
- Optimize CEM by alpha-expansion algorithm and get mask M
- **Inverse transform M to original image**
- Build dual-level CRF with as stated in section 3. Final segmentation is obtained by α-expansion [10] [11] [12].

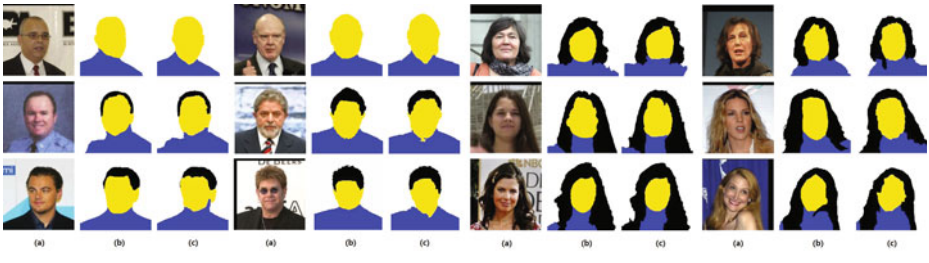


Fig. 6. Examples of Segmentation result. (a) Original Image (b) Manually labeled result (c) Our Segmentation Result. Color code for labels are: white - background, yellow - face, black -hair and blue - clothes.

where $r(j)$ is the rating of the j-th image and C_m is the normalization constant to make that a perfect ordering gets NDCG scores 1. In our experiment, $r(j) = C(P_0, P_j)$. In Figure 7(a), we illustrate our result of $m = 1$ in each location with comparison with a straightforward ranking algorithm using histogram and

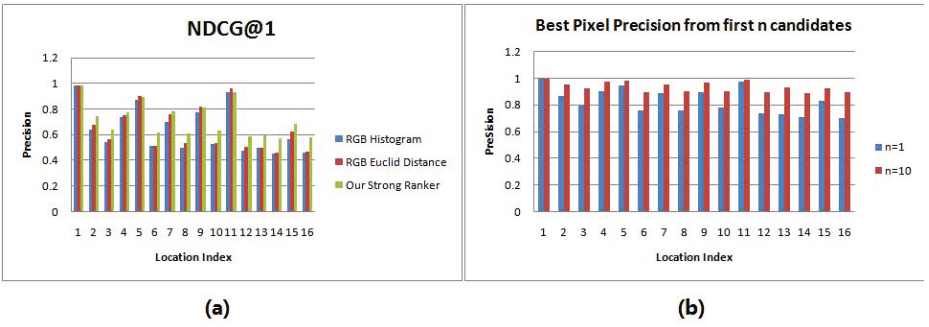


Fig. 7. (a) NDCG at the first image. (b) Best Pixel Precision from the first and the first ten candidate exemplars.

Euclid distance in RGB space. Our algorithm outperforms the straightforward one in almost every location. Especially in the difficult patches, the precision can be improved by 8% to 10%.

And we also test the pixel precision of the best candidate obtained by our strong rankers. Pixel level segmentation accuracy defined as:

$$presion = \frac{\sum_i^n \delta\left(L_i = \tilde{L}_i\right)}{n} \tag{15}$$

where n is the size of the current image. L_i and $\tilde{L}_i$ denotes the label of algorithm result and ground truth of pixel respectively. In Figure 7(b), we show the pixel precision of the best candidate in each location respectively. For most patches, our ranker can find acceptable exemplars for them.

In Figure 8, we give a qualitative CEM example with and without neighborhood consistency constraint. As explained before, independent search for exemplar patch can cause ambiguity inevitably. Neighborhood consistency constraint enforces the continuity between overlapped patches to improve the model robustness for ambiguity.

CEM without neighborhood consistency constraint can achieve a pixel precision of 84.6%. Incorporate the constraint into CEM can improve the precision to 86.3%. As numerous work [15] [16] [17] suggested, incorporating segments prior benefits the segmentation accuracy and robustness. In our problem, the precision of final segmentation result by Dual-Level CRF can reach 89.1%, which outperform Single-Level one by 1.5%. Although they bring only a slight increase in the segmentation accuracy quantitatively, they contribute significantly to subjective quality improvement on segmentation, just as stated in [15], a small increase in the pixel-wise accuracy will actually make a large improvement on the quality of segmentation.

We also test images not included in our manually labeled library. Some of the results are given in Figure 9 to demonstrate its generalization ability. Due to lack

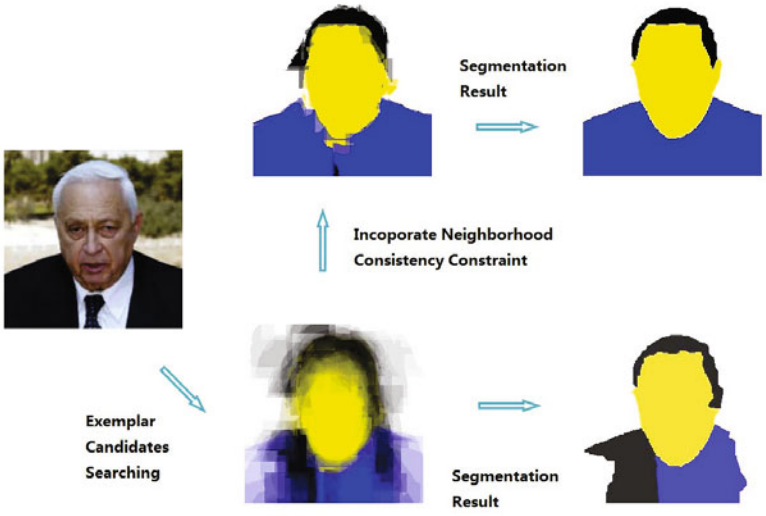


Fig. 8. Comparison between CEM with and without NLC Constraint

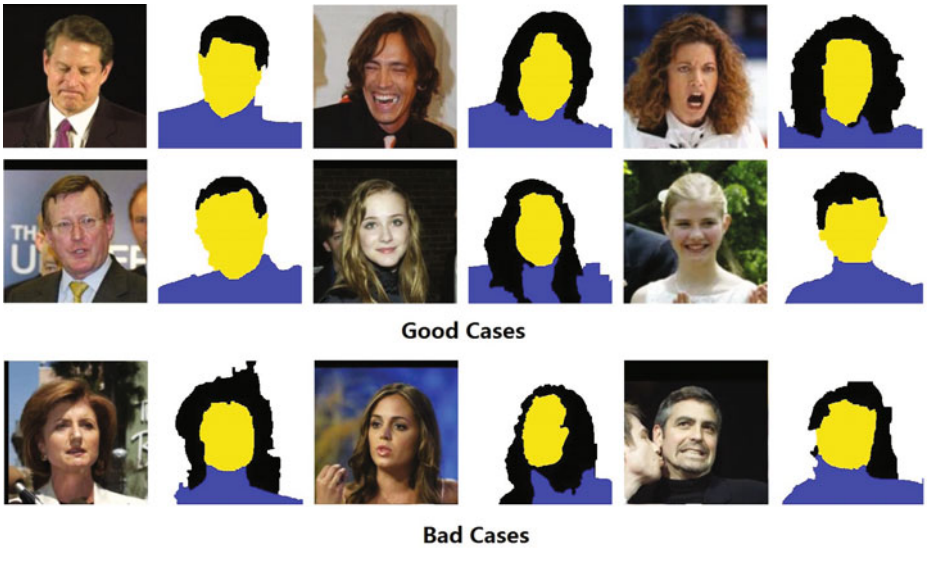


Fig. 9. More Segmentation Results

of technique details of [5], we have not tried to compare with it. Nevertheless we think our method is more powerful in dealing with various hair styles. We tested on 1000 selected face images in front view that are somewhat similar to the exemplars, and about 80% of segmentation results are subjectively acceptable. Unsatisfactory cases occur where hair is confused with background or shadows.

However, if a test image exists in the exemplar library, it will get the exact result. This characteristics guarantees our approach's extensibility since a new hair style can be easily extended by adding its manually labeled result into the exemplar library.

5 Conclusion

In this paper, we propose a novel Compositional Exemplar-based Model for hair style representation and segmentation. CEM generates hair style for the input image in the Divide-and-Conquer manner, which can be divided into the decomposition stage and composition stage naturally. For the decomposition stage, we design a group *Semantic Layout Similarity* features and combine them into a strong ranker by RankBoost algorithm. In the composition stage, we introduce the *Neighbor Label Consistency* Constraint to CEM and define the *consistency function* skillfully to ensure its *regularity*. Final segmentation result is obtained by the inference of Dual-Level Conditional Random Field. Experiment results on face images from Labeled Faces in the Wild data set show its effectiveness. In future, we will try to include side views into the library and speed up the searching procedure.

References

1. Paris, S., Briceo, H.M., Sillion, F.X.: Capture of hair geometry from multiple images. In: SIGGRAPH, Los Angeles, CA, United states, vol. 23, pp. 712–719 (2004)
2. 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: SIGGRAPH, vol. 27 (2008)
3. Ward, K., Bertails, F., Kim, T.Y., Marschner, S.R., Cani, M.P., Lin, M.C.: A survey on hair modeling: Styling, simulation, and rendering. IEEE Transactions on Visualization and Computer Graphics 13, 213–233 (2007)
4. Yacoob, Y., Davis, L.S.: Detection and analysis of hair. PAMI 28, 1164–1169 (2006)
5. chih Lee, K., Anguelov, D., Sumengen, B., Gokturk, S.B.: Markov random field models for hair and face segmentation. In: AFG, Amsterdam, pp. 1–6 (2008)
6. Shotton, J., Winn, J., Rother, C., Criminisi, A.: Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 1–15. Springer, Heidelberg (2006)
7. Borenstein, E., Ullman, S.: Combined top-down/bottom-up segmentation. PAMI 30, 2109–2125 (2007)
8. Wang, X., Tang, X.: Face photo-sketch synthesis and recognition. PAMI 31, 1955–1967 (2009)
9. Jojic, N., Perina, A., Cristani, M., Murino, V., Frey, B.: Stel component analysis: Modeling spatial correlations in image class structure. In: CVPR (2009)
10. Kolmogorov, V., Zabin, R.: What energy functions can be minimized via graph cuts? PAMI 26, 147–159 (2004)
11. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. PAMI 23, 1222–1239 (2001)

12. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. PAMI 26, 1124–1137 (2004)
13. 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)
14. Freund, Y., Iyer, R., Schapire, R.E., Singer, Y.: An efficient boosting algorithm for combining preferences. Journal of Machine Learning Research 4, 933–969 (2004)
15. Kohli, P., Ladick, L., Torr, P.H.S.: Robust higher order potentials for enforcing label consistency. In: CVPR, Anchorage, AK, United states (2008)
16. Larlus, D., Jurie, F.: Combining appearance models and markov random fields for category level object segmentation. In: CVPR, Anchorage, AK, pp. 1–7 (2008)
17. Pantofaru, C., Schmid, C., Hebert, M.: Object recognition by integrating multiple image segmentations. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 481–494. Springer, Heidelberg (2008)
18. Deng, Y., Manjunath, B.S.: Unsupervised segmentation of color-texture regions in images and video. PAMI 23, 800–810 (2001)
19. Huang, G.B., Berg, T., Ramesh, M.: Labeled faces in the wild: A database for studying face recognition in unconstrained environments, University of Massachusetts, Amherst, Technical Report (2007)
20. Huang, C., Ai, H., Li, Y., Lao, S.: High-performance rotation invariant multiview face detection. PAMI 29, 671–686 (2007)
21. Zhang, L., Ai, H., Xin, S., Huang, C., Tsukiji, S., Lao, S.: Robust face alignment based on local texture classifiers. In: ICIP, vol. 2, pp. 354–357 (2005)
22. Jarvelin, K., Kekalainen, J.: Cumulated gain-based evaluation of ir techniques. ACM Transactions on Information Systems 20, 422–446 (2002)

Descriptor Learning Based on Fisher Separation Criterion for Texture Classification

Yimo Guo[1,2], Guoying Zhao[1], Matti Pietikäinen[1], and Zhengguang Xu[2]

[1] Machine Vision Group, Department of Electrical and Information Engineering, University of Oulu, Finland
[2] School of Information Engineering
University of Science and Technology Beijing, China

Abstract. This paper proposes a novel method to deal with the representation issue in texture classification. A learning framework of image descriptor is designed based on the Fisher separation criteria (FSC) to learn most reliable and robust dominant pattern types considering intra-class similarity and inter-class distance. Image structures are thus be described by a new FSC-based learning (FBL) encoding method. Unlike previous handcraft-design encoding methods, such as the LBP and SIFT, supervised learning approach is used to learn an encoder from training samples. We find that such a learning technique can largely improve the discriminative ability and automatically achieve a good tradeoff between discriminative power and efficiency. The commonly used texture descriptor: local binary pattern (LBP) is taken as an example in the paper, so that we then proposed the FBL-LBP descriptor. We benchmark its performance by classifying textures present in the Outex_TC_0012 database for rotation invariant texture classification, KTH-TIPS2 database for material categorization and Columbia-Utrecht (CUReT) database for classification under different views and illuminations. The promising results verify its robustness to image rotation, illumination changes and noise. Furthermore, to validate the generalization to other problems, we extend the application also to face recognition and evaluate the proposed FBL descriptor on the FERET face database. The inspiring results show that this descriptor is highly discriminative.

1 Introduction and Previous Work

Texture is an inherent property of objects and scenes. Texture analysis aims to interpret and understand real-world visual patterns, which would be used in image filtering, classification, segmentation, indexing and synthesis. The general texture classification problem being addressed can be concluded as: given a texture image obtained under certain illumination and viewpoint condition, categorize it as belonging to one of the pre-learned texture classes. In this paper, we will focus on the classification of textures from their appearance taken under varying conditions. This is difficult as changing viewpoint and illumination could have dramatic impacts on the appearance of materials and may lead to large intra-class variation and small inter-class distance.

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 185–198, 2011.

Automatic texture classification has been extensively studied in the past decades. Existing features and techniques vary from image patch exemplars to filter or wavelet based methods [1]. Some representative ones include scale-invariant feature transform (SIFT) and related methods [2,3], local binary patterns (LBPs) and its extensions [4,5,6,7,8,9], texton-based representation methods [10,11] [1], grey level difference or co-occurrence statistics [12], methods based on multi-channel filtering or wavelet decomposition [13,14,15], Gaussian Markov random field (GMRF) and other random field methods [16,17]. Impressively, descriptor based approaches performed surprisingly well in real world situations, such as LBP, SIFT and Histogram of Oriented Gradients (HOG) [18]. No matter they encode relative intensity magnitudes or quantized image gradients, an encoding method or descriptor would be better to get an ideal balance between discriminative ability (meanwhile the robustness against condition variance) and efficiency. However, as these handcrafted descriptors produce unevenly distributed histograms, they would inevitably encounter the problem brought by rarely occurring codes. The resulting histogram might be less informative and less compact, which could degrade the discriminative ability of the image descriptor.

For example, it has been pointed out that LBP, a widely used texture descriptor, using the full set of histogram may not be reliable to describe the input image and yield good classification result because some pattern types rarely happen [7]. Uniform patterns [5], an extension of the LBP, are supposed to represent fundamental images structures, such as edges, flat areas and spots, which are usually dominant patterns among all LBP types (i.e., have proportion above 85%). Using non-dominant LBP histogram bins as image features would lead to severe problems because the histogram might be sparse and many bins might have too few pattern occurrences. However, in some cases, uniform patterns are still not dominant patterns. When texture images have complicated shape and edge type, uniform patterns only occupy a small proportion among all LBP types [9]. As the radius and number of neighboring samples increase, uniform patterns will have a much smaller proportion among all LBP types [5]. Especially, when the number of neighboring samples increases, it is difficult for a particular LBP to match the criteria to become a uniform pattern. Because uniform patterns are defined to have at most two bit-wise transitions across binary digits of each neighboring pixel, the more neighboring samples the center pixel has the more possible transitions there will be. Meanwhile, the number of all possible pattern types will increase faster than that of the possible uniform patterns. In this way, it becomes difficult to cover a significant proportion among all LBPs.

Then the issue becomes whether effective dominant patterns could be learned so that those pattern types which are reliable, robust and highly discriminative can be used for image representation. One recent method is dominant local binary patterns (DLBPs) which extract dominant patterns from the original LBPs by statistics [8]. It was later combined with filter banks and reported a better result than LBP [9]. However, as it calculates the average pattern occurrence of all

images in the training set regardless of intra-class similarity and inter-class differences, the discriminative ability can easily be weakened under varying conditions.

In this paper, we first propose the FSC-based learning framework, then apply it with LBP and present the FBL-LBP descriptor to extract dominant patterns as features for classification. The main contributions of the framework lie in: 1) learning the most reliable and robust dominant pattern types of each class instead of using fixed pattern types; 2) taking both the intra-class similarity and inter-class distance into account in the learning stage, which makes it obtain optimized pattern types according to its particular application; 3) considering dominant pattern type, which is the complementary discriminative information, and pattern type occurrence in image description; 4) being easily generalized by combining with other histogram descriptors for different purposes. The rotation invariance can be implemented by replacing the original histogram with, for example, rotation-invariant LBPs.

2 Texture Image Representation by FBL Descriptor

In this section, we describe the details of the FSC-based learning framework and the feature extraction by FBL-LBP descriptor. The learning framework includes three stages: (1) The learning stage. Determine most reliable dominant types for each class. Then, all the learnt dominant types of each class are merged and form the global dominant types for the whole database; (2) Extract global dominant types learnt in stage (1) of the training set; (3) Extract the global dominant types learnt in stage (1) of the testing set. Finally, features obtained in stages (2) and (3) are served as inputs to the classifier. Each stage will be explained in the following subsections, respectively.

2.1 The Learning Stage

The learning stage of the proposed framework is based on FSC [19,20], which is often used to evaluate the discriminative ability of features. According to the Fisher criterion, the maximum ratio of between-class scatter to within-class scatter leads to the best separation among projected sets. Given a training set containing classes of objects, let the similarities of histograms from different samples of the same class compose the intra-class similarity space. Those samples from different classes compose the extra-class similarity space. The optimal discrimination among data can be obtained by maximizing the sample mean among different classes and, meanwhile, minimizing the intra-class scatter of data. In this way, to learn most reliable and robust dominant pattern types, we carry out FSC in the learning stage by first filtering reliable dominant types from the original histograms for each class to keep the intra-class similarity, and then form the global dominant types by merging dominant types among different classes. LBPs are adopted as the original histograms in this framework as its broad use in texture classification. We will explain how it could be combined with the FSC-based learning framework and obtain the FBL-LBP descriptor.

Supposing a training image set $x_1, x_2, .., x_m$, which belongs to C classes, we have n_c images belonging to class c. Let $\boldsymbol{f}_i$ denote the histogram of all possible LBP types of interest in image i for given radius R and neighboring samples N. If rotation invariant property is required, the framework should consider all possible rotation invariant LBP types in this step. Each LBP type is characterized by the general LBP type label, defined as Equation 1:

$$LBP_{N,R} = \sum_{l=0}^{N-1} u(t_l - t_c)2^l, \tag{1}$$

where $u(x)$ is the step function with $u(x) = 1$ if $x \geq 0$ and $u(x) = 0$ otherwise. t_l denotes the intensity of neighboring pixel l, and t_c denotes the intensity of the center pixel. When the rotation invariant property is required, LBP labels can be calculated by Equation 2:

$$LBP_{N,R} = \min_{0 \leq d < N} \left(\sum_{l=0}^{N-1} u(t_l - t_c)2^{[(l+d) mod\ N]} \right). \tag{2}$$

Let p denote the total possible number of LBP types of interest and $f_{i,j}$ denote the number of occurrences of pattern type j in image x_i. We define the set of dominant LBPs of each image as the following definition.

Definition 1: *Dominant LBP set of an image is the minimum set of LBP types which can cover n% of all LBPs of the image.*
Definition 1 is expressed using Equation 3 in order to find a set J_i for image x_i (i=1,...,m), which can be implemented by Algorithm 1:

$$J_i = \arg\min_{|J_i|} \left(\frac{\sum_{j \in J_i} f_{i,j}}{\sum_{k=1}^{p} f_{i,k}} \right) \geq n\%, \tag{3}$$

where p is the total number of all possible local binary pattern types and $|J_i|$ denotes the number of elements in set J_i ($J_i \subseteq [1, 2, ..., p]$).

Based on the FSC, the most discriminant features should have large inter-class mean distance and small intra-class variation. Thus, to learn reliable dominant LBP set of each class, we remove the outlier caused by noise or illumination variation for individual images in the same class, only considering the common features. This is reflected by Fig. 1. It is also shown that not only pattern types but also the number of dominant LBP set elements of each image belonging to the same class might be changed as illumination changes or due to other distortion factors. If we consider all possible pattern types that belong to dominant LBP set of each sample, the image description will be not robust and stable enough to characterize the whole class. Therefore, only the pattern types that consistently belong to dominant pattern type sets of each image in this class are adopted as the dominant pattern type set of this class. The procedure is described by Algorithm 2.

After the learning of most reliable dominant pattern set of each class c (c =1,...,C), we construct the global dominant pattern set of interest for the whole

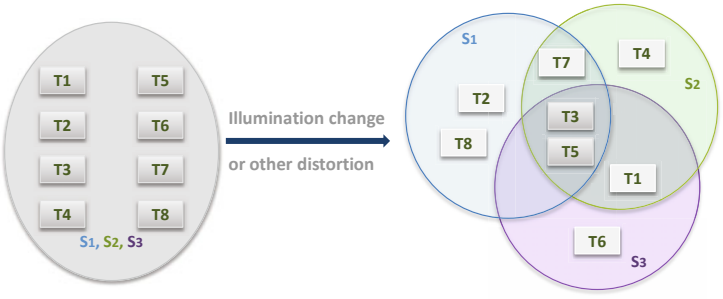


Fig. 1. The left circle denotes the ideal situation: three samples (i.e., S_1, S_2, S_3) belonging to the same class should have the same set of dominant LBPs (denoted by TN, where N is the pattern type label). The right circles denote the resulting dominant pattern types of each sample after distortion. Some dominant pattern types are changed because of imaging conditions. The number of the dominant LBP set elements of each sample might be different. In this example, only the pattern types T3 and T5 remain as dominant patterns for all samples after distortion, which would construct the most reliable dominant pattern set of this class.

database using Algorithm 3. In this stage, pattern occurrences of pattern types are considered in J_{global} instead of using fixed pattern set as in conventional methods, e.g., the uniform LBP. This has at least two advantages. First, pattern types in J_{global} are more reliable to characterize the property of each class, as only the pattern types that consistently belong to the dominant pattern set of each sample are preserved for each class. Second, J_{global} is guaranteed to be able to cover all dominant patterns across different classes, as it is the union of the most reliable dominant patterns of each class. In following stages, each element of J_{global} represents a pattern type of interest whose frequency occurrence will be calculated.

2.2 The Training Stage

Given the global dominant pattern set of interest J_{global} obtained in the learning stage, we extract occurrence histogram of pattern types in J_{global} as features for each image. Then image x_i, belonging to training set S_{train}, can be represented by feature vector $\boldsymbol{y}_i$, which not only encodes the occurrence frequency of each dominant pattern type, but also considers pattern type information. Each dimension of $\boldsymbol{y}_i$ represents a particular fixed type of dominant pattern and these dominant pattern types also contain discriminative information as the pattern occurrence, which makes the proposed feature more powerful in classification.

2.3 The Testing Stage

In the testing stage, the dominant LBP histogram is calculated for each testing image based on J_{global} similar to the procedure performed on the training set.

Algorithm 1. Find the dominant LBP set of an input image x_i

Input: The original histogram $\boldsymbol{f}_i$ of x_i for all LBP types of interest.
Output: The dominant LBP set J_i of image x_i.
1. Initialize a reference pattern type record vector $\boldsymbol{V}$ where $V[i]=(i-1)$ (i=1,...,p).
2. Sort $\boldsymbol{f}_i$ in descending order, resulting in a new histogram $\hat{\boldsymbol{f}}_i$. Change the configuration of $\boldsymbol{V}$ according to the element switching order from $\boldsymbol{f}_i$ to $\hat{\boldsymbol{f}}_i$, resulting in a new vector $\hat{\boldsymbol{V}}$. Now the top h entries of $\hat{\boldsymbol{f}}_i$ denote the occurrence frequencies of the top h most dominant patterns and the top h entries of $\hat{\boldsymbol{V}}$ record the pattern labels of the top h most dominant patterns.
3. FOR k = 1 to p
   IF $\left(\left(\frac{\sum_{l=1}^{k}\hat{f}_{i,l}}{\sum_{l=1}^{p}\hat{f}_{i,l}}\right) \geq n\%\right)$
   BREAK;
   END IF
   END FOR
4. $J_i = \{\hat{V}[1], ..., \hat{V}[k]\}$
5. Return J_i

Algorithm 2. Find the dominant LBP set of class c

Input: n_c input training images belonging to class c.
Output: The dominant LBP set JC_c of class c.
1. Calculate the dominant LBP set J_1 of the first image belonging to class c, and initialize $JC_c = J_1$.
2. FOR each image $i = 2$ to n_c belonging to class c
   Calculate its dominant LBP set J_i by Algorithm 1.
   $JC_c = JC_c \cap J_i$.
   END FOR
3. Return JC_c.

The learning-based LBPs extracted from the training and testing set will be finally served as inputs to classifier for classification. The pipeline of the FSC-based learning framework is shown in Fig. 2.

3 Experimental Design and Results

We test the performance of the proposed method for texture classification on the Outex_TC_0012 database [21], KTH-TIPS2 database [22] and Columbia-Utrecht (CUReT) database [23] in three different scenarios: rotation invariant texture classification, material categorization, and texture classification under variant imaging conditions. The proposed descriptor is compared against the non-invariant uniform local binary pattern LBP^{u2} (or the rotation invariant version LBP^{riu2}) and DLBP on all these databases. Some well-known methods are also compared with on some of these databases. The rotation invariant LBP is adopted as the original histogram of the framework for all texture classification

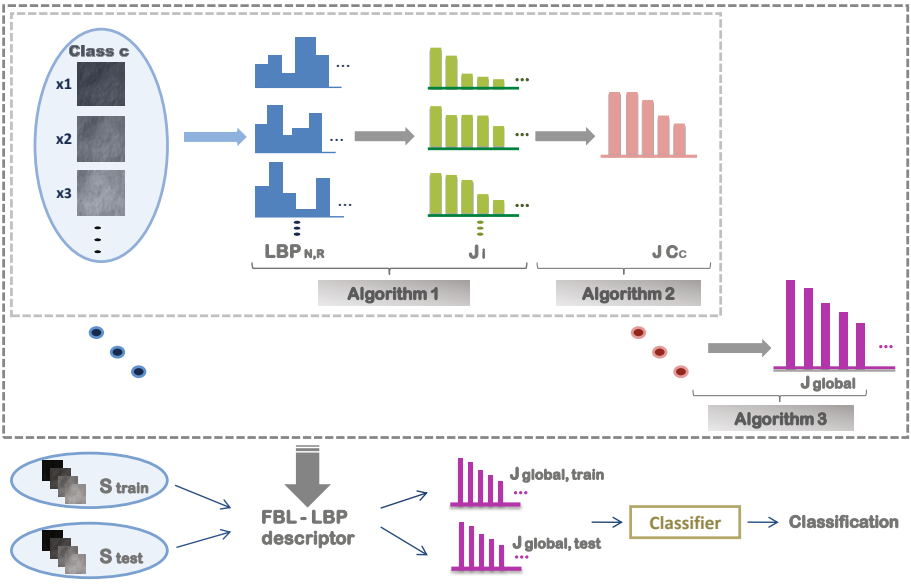


Fig. 2. The FSC-based learning framework

tasks in this paper and the threshold is set to be 90%. The descriptor is further evaluated on the FERET face database [24] to prove its ability in Biometrics.

3.1 Rotation Invariant Texture Classification

We use the Outex_TC_0012 database to test rotation invariant texture classification methods. This database consists of 9120 images representing 24 different textures imaged under different rotations and lightings. The test set contains 20 training images for each texture class. The training images are under single orientation whereas different orientations are present in the total of 8640 testing images. The total classification rates over all test images are listed in Table 1, which are derived from the setup by using the nearest neighbor (NN) classifier.

Algorithm 3. Construct the global dominant pattern set

1 *Input: The dominant LBP set* JC_c $(c = 1,\ldots,C)$ *of each class obtained by Algorithm 2.*

2 *Output: The global dominant pattern set* J_{global}.

1. Initialize $J_{global} = \emptyset$.
2. FOR $i = 1$ to C
   $J_{global} = J_{global} \cup JC_i$
   END FOR
3. Return J_{global}.

It can be observed at all tested scales that rotation invariant features LBP^{riu2}, LBP-HF, DLBP and FBL-LBP provide higher classification rates than non-invariant feature LBP^{u2}. The performance of the new features is clearly better than that of the LBP^{riu2}. This improvement demonstrates its effectiveness in feature extraction. As the number of neighboring samples increases, the total number of possible LBPs will dramatically increase. In this case, many patterns will be produced with low occurrence frequencies and the pattern histogram becomes sparse, which makes the image representation unstable. The FBL framework solves this problem by considering only the most dominant patterns and eliminating unreliable patterns to reduce negative effects.

LBP-HF is cited as one representative method combining the LBP with pattern transform. FBL-LBP performs better than it at the scales (24,3) and (16,2)+(24,3). In addition, the proposed descriptor is compared with the recent DLBP, which is also a learning-based method. To be specific, in this paper, we adopt the nearest neighbor classifier for DLBP and do not use any preprocessing prior to feature extraction. The best result in Table 1 is achieved by our method at the scale (8,1)+(16,2)+(24,3). For fair comparison purpose, the dominant pattern threshold of DLBP is set to 90%, which is the same as FBL-LBP For further comparison, we refer to the MR8, a filter bank based texton method [11], which got 76.1% on this database [7], but not listed here.

Table 1. Texture classification rates on Outex_TC_0012 dataset

Parameters	LBP^{u2} [7]	LBP^{riu2} [7]	LBP-HF [7]	DLBP [9]	FBL-LBP
(8,1)	0.566	0.646	0.773	0.560	0.691
(16,2)	0.578	0.791	0.873	0.687	0.825
(24,3)	0.450	0.833	0.896	0.754	0.901
(8,1)+(16,2)	0.595	0.821	0.894	0.778	0.833
(8,1)+(24,3)	0.512	0.883	0.917	0.820	0.905
(16,2)+(24,3)	0.513	0.857	0.915	0.837	0.927
(8,1)+(16,2)+(24,3)	0.539	0.870	0.925	0.849	**0.928**

3.2 Material Categorization

Image descriptors are tested on the KTH-TIPS2 database [22] for material categorization. This database contains four samples of 11 different materials, each sample imaged at nine different scales and 12 lighting and pose setups, totaling 4572 images. The NN classifier is trained with one sample (i.e. 9×12 images) per material category. The remaining $3 \times 9 \times 12$ images are used for testing. This is repeated with 10000 random combinations as training and testing data and the mean categorization rate over the permutations is used to assess the performance.

Results of the LBP^{u2}, LBP^{riu2}, LBP-HF and DLBP are listed in Table 2. It can be observed that FBL-LBP has obvious superiority compared to other methods in all cases. This is most likely that abundant orientations are present in the training data for learning. Similarly, the performance of LBP^{riu2} is consistently lower probably because different orientations are contained in training samples so rotational invariance does not benefit much [7]. The multi-resolution (8,1)+(16,2)+(24,3) is able to give a good result but not the best as the scale (24,3) does not work well, which also happens to other methods at this scale. This might be brought by the scale variation properties of this database. However, FBL-LBP with the scale (8,1)+(16,2) achieves a slight improvement over it possibly as more discriminative information is contained within smaller radius.

3.3 Texture Classification under Variant Imaging Conditions

The CUReT database contains images of 61 materials and includes many surfaces commonly seen in our environment [23]. Each of the materials in the database has been imaged under 205 different viewing and illumination conditions. The effects of surface normal variations such as specularities, reflections and shadowing are evident. This database also includes some man-made textures, and is highlighted due to abundant imaging conditions. These make it far more challenging and become a benchmark widely used to assess classification performance.

The experiments are conducted on the CUReT database in the same way as in [1]. The cropped database has a total of 5612 images. Out of these, 46 images per class are randomly chosen for training and the remaining 46 per class are chosen for testing. The cropped CUReT database can be downloaded from [25]. Table 3 lists the best classification rates of different features. For comparison, results obtained by LBP^{riu2}, texton based representation method and DLBP are presented. LBP^{riu2} and FBL-LBP follow the same setting (8,1)+(8,3)+(8,5), and DLBP is set to (16,2) with its best results. It is observed that FBL-LBP can even achieve a slightly higher classification rate 97.61% than 97.47% obtained by texton method when training samples are chosen randomly for 61 class problems.

From experimental results conducted on the three main challenging texture databases, the proposed descriptor is shown to be stable and discriminative enough to represent texture images.

3.4 Face Recognition

The last experiment is performed to assess whether FBL-LBP could provide effective representation in face recognition. The experiment is conducted on the FERET face database following the standard FERET protocol in [24] which is more challenging than the one used in [6] as less training images are available.

For the FERET database, we use Fa as gallery, which contains 1196 frontal images of 1196 subjects. The probe sets consist of Fb, Fc, Dup I and Dup II. Fb contains 1195 images of expression variations, Fc contains 194 images taken under different illumination conditions, Dup I has 722 images taken later in time and Dup II (a subset of Dup I) has 234 images taken at least one year after the

Table 2. Comparison of classification results for material categorizations

Parameters	LBP^{u2} [7]	LBP^{riu2} [7]	LBP-HF [7]	DLBP [9]	FBL-LBP
(8,1)	0.528	0.482	0.525	0.458	0.619
(16,2)	0.511	0.494	0.533	0.460	0.624
(24,3)	0.502	0.481	0.513	0.459	0.609
(8,1)+(16,2)	0.536	0.502	0.542	0.456	**0.631**
(8,1)+(24,3)	0.542	0.507	0.542	0.468	0.613
(16,2)+(24,3)	0.514	0.508	0.539	0.458	0.623
(8,1)+(16,2)+(24,3)	0.536	0.514	0.546	0.461	0.626

Table 3. Comparison of classification results on CUReT database

LBP^{riu2} [5]	Texton method [1]	DLBP [9]	FBL-LBP
0.9624	0.9747	0.9593	**0.9761**

corresponding Gallery images. Using Fa as the gallery, we design the following experiments: (i) use Fb as probe set to test the efficiency of the method against facial expression; (ii) use Fc as probe set to test the efficiency of the method against illumination variation; (iii) use Dup I as probe set to test the efficiency of the method against short time; (iv) use Dup II as probe set to test the efficiency of the method against longer time. All images in the database are cropped and normalized to the resolution of 128×128 using eye coordinates provided. They are uniformly divided into 7×7 non-overlapping sub-regions.

The feature extraction using FBL-LBP for face recognition includes these procedures: (1) divide each image from the training set uniformly into 7×7 sub-regions. Global dominant pattern sets are constructed for each region and then connected to be the overall dominant types for the whole database. (2) calculate LBP histogram of dominant types for the training set and testing set, which will be served as inputs to classifier. The recognition rates of different methods are listed in Table 4. We use the same setting as [6]: eight neighboring samples, radius two and the same block weights. The result is lower than that in [6] as a more strict protocol is adopted. For FBL-LBP, we also use the (8,2) setting and normal patterns as the original histogram with threshold 70%, while DLBP follows (16,2) setting with its best result on this database. The threshold has a different value from the one used in texture classifications considering that the number of classes and intra-class variations are higher in face database. As the number of classes increases, more dominant pattern types are needed, so that more patterns are used to construct dominant pattern histograms. However, the dominant pattern proportions of all original patterns in this experiment is just 34.77%, which means the learning-based method does not degrade with this threshold.

Table 4. The recognition rates on the FERET database probe set

Methods	Fb	Fc	Dup I	Dup II
PCA [27]	0.749	0.113	0.302	0.081
LBP^{u2} [6]	0.874	**0.572**	0.389	0.385
DLBP [9]	0.881	0.516	0.362	0.349
FBL-LBP	**0.899**	0.536	**0.449**	**0.389**

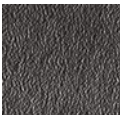
From the comparison, FBL-LBP has a better performance on the Fb, Dup I and Dup II than that of the uniform LBP, especially on Dup I. Although the classification rate is not the best on Fc, it is still higher than most other face recognition methods in [26], which follow the same protocol. FBL-LBP does not perform significantly better probably because the condition variations needed are not present enough in the training data, which could influence the performance of learning-based method.

4 Discussion

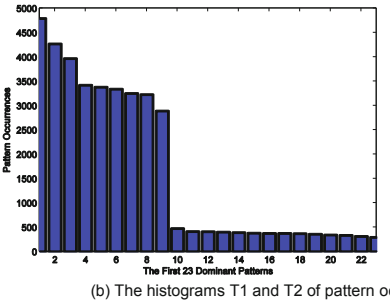
In this section, we will discuss how the FBL-LBP retains the discriminativeness of the original histogram and its relationships with DLBP, uniform LBP and texton-based approaches.

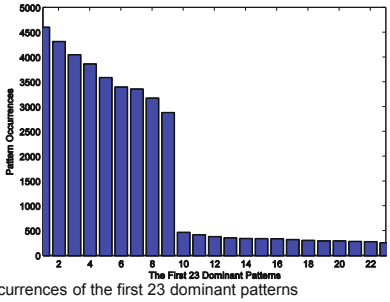
The concern is that previous methods mainly take pattern occurrence into account in image representation, which may not be able to provide enough discriminative patterns for texture classification. As illustrated in Fig. 3(a), there are two texture images G1 and G2 from the CUReT database belonging to two classes. We extract patterns using the DLBP (N = 8, R = 1, proportion = 90%). F1 and F2 are pattern occurrences of their first 23 dominant patterns in descending order, which are very similar to each other. Their histograms T1 and T2 are given for comparison in Fig. 3(b). The labels L1 and L2 listed in Fig. 3(c) are the dominant pattern types of F1 and F2 for each entry, respectively. Their corresponding dominant pattern types are obviously different from each other. In this case, it becomes difficult to classify them just using the pattern type occurrence. But it becomes possible when adding dominant pattern type information as we proposed in image representation, as shown in Fig. 4, the patterns that are obtained using the proposed framework (N = 8, R = 1, threshold = 90%). Moreover, considering intra-class similarity and inter-class distance of the database, FBL-LBP uses the intra-class intersection and inter-class unit as statistics to extract dominant patterns from the original histogram, instead of calculating the average pattern occurrence on the whole training set as in [9], from which its performance could further benefit in the texture classification.

In order to prove the ability in extracting discriminative patterns, we explore the relationship between the FBL-LBP and widely used uniform LBP by calculating the average uniform pattern proportions of FBL-LBPs, as listed in Table 5. The recognition rates of FBL-LBP remain higher even with less intersection with the uniform LBP, which shows it could capture effective patterns from the




(a) Two texture images G1 and G2 belonging to two classes





(b) The histograms T1 and T2 of pattern occurrences of the first 23 dominant patterns

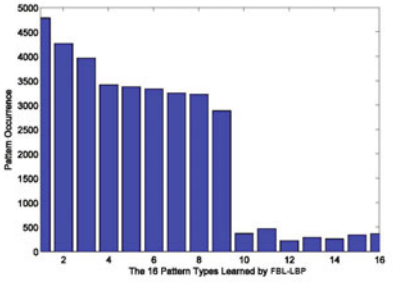
L1	1	34	35	8	27	4	0	16	2	32	5	21
L2	4	8	16	35	2	34	1	27	0	31	3	14
L1	7	24	26	3	6	25	31	18	14	13	10	
L2	24	18	13	10	25	7	6	15	12	21	26	

(c) Corresponding pattern type of each entry

Fig. 3. The dominant patterns produced by DLBP

Table 5. The average uniform pattern proportions of all FBL-LBPs (n=90)

Outex_TC_0012				KTH-TIPS			
(N,R)	Proportions	LBP^{riu2}	FBL-LBP	(N,R)	Proportions	LBP^{riu2}	FBL-LBP
(8,1)	88.13%	0.646	0.691	(8,1)	82.68%	0.482	0.691
(16,2)	67.71%	0.791	0.825	(16,2)	58.91%	0.494	0.624
(24,3)	42.97%	0.833	0.901	(24,3)	29.83%	0.481	0.609



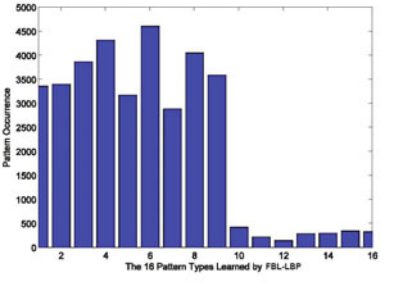


Fig. 4. The dominant patterns produced by FBL-LBP

original histogram discarded by uniform LBP. Especially, when the number of neighboring samples increases, non-uniform patterns could be effective for image representation. We would suppose when the number of the neighboring samples and radius increase, FBL-LBPs performs even better comparing to the uniform patterns, since the former can take non-uniform patterns that are in dominant set but discarded, while the latter are not that dominant as with smaller number of neighboring samples and radius.

5 Conclusions

In this paper, we propose a descriptor learning framework for texture classification. The framework is based on FSC to learn most reliable and robust dominant pattern types considering intra-class similarity and inter-class distance. LBP was taken as an input to this framework for texture classification and face recognition. Non-dominant LBP histogram would lead to severe problems caused by sparse histogram, however, it is shown that FBL-LBP, the descriptor obtained by combining the proposed framework with LBP, can retain dominant patterns and eliminate unreliable patterns to reduce negative effects. FBL-LBP differs from previous LBP approaches since FBL framework learns robust dominant types of each class instead of using fixed pattern types. To get reliable patterns adaptive to particular applications, the learning process takes intra-class similarity and inter-class distance into account. To strengthen the discriminativeness of image description, dominant pattern type is adopted as a complement to the pattern type occurrence. In addition, this framework is easy to generalize for other purposes by introducing different histogram descriptors.

Acknowledgements. The research was sponsored by the Infotech Oulu Graduate School and the Academy of Finland.

References

1. Varma, M., Zisserman, A.: A statistical approach to material classification using image patch exemplars. IEEE Transactions on Pattern Analysis and Machine Intelligence 31, 2032–2047 (2009)
2. Lowe, D.: Object recognition from local scale-invariant features. In: International Conference on Computer Vision, vol. 2, pp. 1150–1157 (1999)
3. Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: a comprehensive study. International Journal of Computer Vision 73, 213–238 (2007)
4. Ojala, T., Pietikäinen, M., Harwood, D.: A comparative study of texture measures with classification based on feature distributions. Pattern Recognition 29, 51–59 (1996)
5. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 971–987 (2002)
6. Ahonen, T., Hadid, A., Pietikäinen, M.: Face recognition with local binary patterns. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 469–481. Springer, Heidelberg (2004)

7. Ahonen, T., Matas, J., He, C., Pietikäinen, M.: Rotation invariant image description with local binary pattern histogram fourier features. In: Salberg, A.-B., Hardeberg, J.Y., Jenssen, R. (eds.) SCIA 2009. LNCS, vol. 5575, pp. 61–70. Springer, Heidelberg (2009)
8. Liao, S., Chung, C.S.: Texture classification by using advanced local binary patterns and spatial distribution of dominant patterns. In: International Conference on Acoustics, Speech, and Signal Processing, vol. 1, pp. 1221–1224 (2007)
9. Liao, S., Law, M., Chung, C.S.: Dominant local binary patterns for texture classification. IEEE Transactions on Image Processing 18, 1107–1118 (2009)
10. Schaffalitzky, F., Zisserman, A.: Viewpoint invariant texture matching and wide baseline stereo. In: International Conference on Computer Vision, vol. 2, pp. 636–643 (2001)
11. Varma, M., Zisserman, A.: A statistical approach to texture classification from single images. International Journal of Computer Vision 62, 61–81 (2005)
12. Weszka, J., Dyer, C.R., Rosenfeld, A.: A comparative study of texture measures for terrain classification. IEEE Transactions on Systems, Man, and Cybernetics 6, 269–285 (1976)
13. Randen, T., Husoy, J.H.: Filtering for texture classification: A comparative study. IEEE Transactions on Pattern Analysis and Machine Intelligence 21, 291–310 (1999)
14. Mallat, S.G.: A theory for multiresolution signal decomposition: The wavelet representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 11, 674–693 (1989)
15. Unser, M.: Texture classification and segmentation using wavelet frames. IEEE Transactions on Image Processing 11, 1549–1560 (1995)
16. Chellappa, R., Chatterjeey, S.: Classification of textures using gaussian markov random fields. IEEE Transactions on Acoustics, Speech, and Signal Processing 33, 959–963 (1985)
17. Cross, G.R.: Markov random field texture models. Ph.D. dissertation, East Lansing, MI (1980)
18. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: International Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 886–893 (2005)
19. Fisher, A.: The mathematical theory of probabilities. Macmillan, Basingstoke (1923)
20. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. The Journal of Machine Learning Research 3, 1157–1182 (2003)
21. Ojala, T., Mäenpää, T., Pietikäinen, M., Viertola, J., Kyllönen, J., Huovinen, S.: Outex-new framework for empirical evaluation of texture analysis algorith. In: International Conference on Pattern Recognition, vol. 1, pp. 701–706 (2002)
22. Caputo, B., Hayman, E., Mallikarjuna, P.: Class-specific material categorisation. In: International Conference on Computer Vision, vol. 2, pp. 1597–1604 (2005)
23. Dana, K.J., van Ginneken, B., Nayar, S.K., Koenderink, J.J.: Reflectance and texture of real world surfaces. ACM Transactions on Graphics 18, 1–34 (1999)
24. Phillips, P.J., Wechsler, H., Huang, J., Rauss, P.: The feret database and evaluation procedure for face recognition algorithms. Image and Vision Computing 16, 295–306 (1998)
25. http://www.robots.ox.ac.uk/vgg/research/texclass/index.html
26. http://www.itl.nist.gov/iad/humanid/feret/perf/score_cms/score_cms.html
27. Turk, M., Pentland, A.: Eigenfaces for recognition. Journal of Cognitive Neuroscience 3, 71–86 (1991)

Semi-supervised Neighborhood Preserving Discriminant Embedding: A Semi-supervised Subspace Learning Algorithm

Maryam Mehdizadeh[1], Cara MacNish[1], R. Nazim Khan[2], and Mohammed Bennamoun[1]

[1] Department of Computer Science and Software Engineering, University of Western Australia
[2] Department of Mathematics of the University of Western Australia

Abstract. Over the last decade, supervised and unsupervised subspace learning methods, such as LDA and NPE, have been applied for face recognition. In real life applications, besides unlabeled image data, prior knowledge in the form of labeled data is also available, and can be incorporated in subspace learning algorithm resulting in improved performance. In this paper, we propose a subspace learning method based on semi-supervised neighborhood preserving discriminant learning, which we call Semi-supervised Neighborhood Preserving Discriminant Embedding (SNPDE). The method preserves the local neighborhood structure of face manifold using NPE, and maximizes the separability of different classes using LDA. Experimental results on two face databases demonstrate the effectiveness of the proposed method.

1 Introduction

Biometric face data are data of high dimensions and are susceptible to the well-known problem of the *curse of dimensionality* when using machine learning techniques. A common approach is to transform the high dimensional data into a lower dimensional subspace which preserves the perceptually meaningful structure of these images. Fisherface [1], and NPEface [2] are two face subspace learning methods. Fisherface, which is a supervised method based on LDA [3], projects the data points along the directions with optimal class separability, and performs subspace learning based on global Euclidean properties of the image data. NPE on the other hand, is an unsupervised subspace learning method, which performs subspace learning based on local neighborhood properties of the high dimensional image data. In this method, an image is considered as a high dimensional vector, that is, a point in a high dimensional vector space, and the set of all faces are assumed to lie on or near a lower dimensional manifold. The aim of NPE is to discover this manifold structure and perform subspace learning with the objective of best preserving the manifold structure.

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 199–212, 2011.

The assumption of NPE is that nearby points share class information, and recognition of points are based on their closest neighbors in the reduced face subspace. However, in face recognition, variability in illumination and expression makes it hard to discern identities based solely on similarity of images. In other words, images in a small neighborhood might belong to different identities. Therefore, in addition to the neighborhood preserving criteria, there is also a need for discriminant analysis of data, so that the projection of two similar images that belong to different identities is not close in the reduced subspace.

In recent years graph-based subspace learning methods have been studied, which encode discriminant information or manifold structure of image data as graphs and perform subspace learning based on *graph preserving criterion*. Graph Embedding (GE) [4] was introduced as a general framework for dimensionality reduction enabling popular methods of subspace learning to be interpreted and implemented as graph based methods. In addition, Cai et al. [5] provided a general framework for subspace learning, and discussed the possibility of constructing multiple graphs to learn the intrinsic discriminant structure of the image data. In addition, they showed that their framework follows the GE view of subspace learning.

In this paper, along the framework introduced by Cai et al. [5] for content-based image retrieval, we propose a semi-supervised subspace learning method for face recognition which uses two graphs that are constructed to encode the necessary information of image data. We call this Semi-supervised Neighborhood Preserving Discriminant Embedding (SNPDE) for face representation and recognition. Our method is constructed based on: (i) graph view of NPE, which builds an adjacency graph that best reflects the geometry of the face manifold; and (ii) graph view of LDA, which builds a graph with edge weights that reflect the discriminant structure of data. The projection function then consists of a set of basis vectors obtained based on a unified objective function incorporating the graph preserving of NPE and LDA. Since SNPDE combines the objective of NPE with discriminative objective of LDA, it is expected to perform better than NPE for face recognition, and this is demonstrated in our results section.

The rest of the paper is organized as follows. In Section 2, We review GE view of subspace learning and discuss the graph view of NPE and LDA. The SNPDE method is described in Section 3. The experimental results are discussed and compared with other methods in Section 4, followed by concluding remarks in Section 5.

2 Graph Embedding and Graph Based NPE and LDA

2.1 Graph Embedding View of Subspace Learning

A given set $\{\mathbf{x_i}\}_{i=1}^{N} \subset \mathbf{R}^n$ of N images can be represented as an image matrix $X = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N]$. The essential task of subspace learning is to find an optimal mapping function that projects the high dimensional face data into a lower dimensional face space $Y = \{\mathbf{y_i}\}_{i=1}^{N} \subset R^d$,where $d << n$. That is,

$$Y = \mathbf{A}^T X \tag{1}$$

where A is an $n \times d$ matrix consisting of a set of basis vectors $\mathbf{a}^* = [\mathbf{a}_1, \ldots, \mathbf{a}_d]$, (where $\mathbf{a}_i \in \mathbf{R}^n$ for $i = 1, \ldots, d$). In graph based subspace learning methods, the data vector is represented as a graph G, such that vertex i of the graph represents vector $\mathbf{x}_i$. An $n \times n$ weight matrix W is then defined such that each edge weight W_{ij} reflects the relationship between data points $\mathbf{x}_i$ and $\mathbf{x}_j$. The objective of Graph Embedding (GE) is to represent each vertex of a graph as a low dimensional vector where the relationship between vertex pairs (i, j) is best preserved.

The GE formulation of subspace learning is as follows

$$\mathbf{a}^* = \underset{a:\ y^T D y=1}{\arg\min} \sum_{i \neq j} (\mathbf{a}^T \mathbf{x}_i - \mathbf{a}^T \mathbf{x}_j)^2 W_{ij} \tag{2}$$

$$= \underset{a:\ y^T D y=1}{\arg\min}\ \mathbf{a}^T X L X^T \mathbf{a} \tag{3}$$

where $L = D - W$ is the *graph Laplacian* [6], and D is a diagonal matrix whose entries are column (or, since W is symmetric, row) sums of W. The optimization in (3) can also be written as

$$\mathbf{a}^* = \underset{a}{\arg\min} \frac{\mathbf{a}^T X L X^T \mathbf{a}}{\mathbf{a}^T X D X^T \mathbf{a}} \tag{4}$$

which reduces to solving the general eigenvalue problem

$$X L X^T \mathbf{a} = \lambda X D X^T \mathbf{a}, \tag{5}$$

or equivalently as

$$\mathbf{a}^* = \underset{a}{\arg\max} \frac{\mathbf{a}^T X W X^T \mathbf{a}}{\mathbf{a}^T X D X^T \mathbf{a}}, \tag{6}$$

which reduces to solving the general eigenvalue problem

$$X W X^T \mathbf{a} = \lambda X D X^T \mathbf{a}. \tag{7}$$

In sections 2.2 and 2.3, we will show that graph view of NPE and LDA can be derived from the GE formulations in (4) and (6).

2.2 Graph View of NPE

In this section, we will review the NPE algorithm and discuss the graph G^{NPE} which can model the learnt manifold structure based on NPE algorithm. We will also reformulate the NPE method based on the GE formulation of subspace learning in (4), and this will help us develop our SNPDE algorithm.

NPE [2] is an unsupervised subspace learning method that inherits the local linear but global nonlinear learning characteristics of Locally Linear Embedding (LLE) [7], a well-known nonlinear dimensionality reduction method. Unlike LLE, which is only applicable to input training data, NPE obtains a linear mapping function based on the training data, that is also applicable to unseen test data. NPE assumes data to lies on a nonlinear manifold and obtains its linear mapping with the aim that the local neighborhood characteristics of the manifold are best preserved. Similar to LLE, NPE characterizes the local neighborhood structure of each data point by linear coefficients W_{ij} such that each data point $\mathbf{x}_i$ can be (approximately) reconstructed from its k-neighbors $\{\mathbf{x}_j\}_{j=1}^{k}$ by $\hat{\mathbf{x}}_i = \sum_{j=1}^{k} W_{ij}\mathbf{x}_j$. NPE then obtains a linear mapping function such that the local linear characteristics identified by W_{ij} are best preserved in the lower dimensional subspace. The actual computations for obtaining the linear mapping by NPE involve solving a generalized eigenvalue problem derived from the cost function of LLE.

The computations for NPE can be divided into the following three steps.

1. **Construct the neighborhood graph**
   The neighborhood weights are obtained by the following optimization:

$$\min \sum_i \left\| \mathbf{x}_i - \sum_j W_{ij}\mathbf{x}_{N(j)} \right\|^2 \tag{8}$$

   subject to the constraints:

$$\sum_{j=1}^{k} W_{ij} = 1, \text{for each } j = 1, \ldots, k. \tag{9}$$

   The neighborhood weight matrix W forms a $k \times n$ matrix, where k is the number of neighboring points for each image data and n is the number of data points in the image database. The $n \times n$ weight matrix W^{NPE} of graph G^{NPE} is obtained as

$$W_{ij}^{NPE} = \begin{cases} (W + W^T - W^T W)_{ij} & \text{if i} \neq \text{j} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

   The structure of the similarity weight matrix W_{ij}^{NPE} was first introduced in [4].

2. **Compute the optimal linear projections**
He et al. [2] obtained the neighborhood preserving mapping of NPE, that is, they obtained the matrix A such that the mapping $X \rightarrow Y$ where $Y = A^T X$ preserves the neighborhood characteristics of the data manifold. Thus similar to (8), we select $\mathbf{y}^* = [\mathbf{y}_1, \ldots, \mathbf{y}_n]$ as:

$$\mathbf{y}^* = \arg\min_{y} \sum_i \left\| \mathbf{y}_i - \sum_j W_{ij} \mathbf{y}_{N(j)} \right\|^2 \tag{11}$$

$$= \arg\min_{y:\ y^T y=1} \mathbf{y}^T (I - W)^T (I - W) \mathbf{y} \tag{12}$$

$$= \arg\min_{a:\ a^T X X^T a=1} \mathbf{a}^T X (I - W)^T (I - W) X^T \mathbf{a} \tag{13}$$

$$= \arg\min_{a:\ a^T X X^T a=1} \mathbf{a}^T X M X^T \mathbf{a} \tag{14}$$

Here I is the $n \times n$ identity matrix and M is the $n \times n$ matrix given by

$$M = (I - W)^T (I - W). \tag{15}$$

The optimal linear projections in (14) are the eigenvectors associated with smallest eigenvalues of the generalized eigenvalue problem

$$X M X^T \mathbf{a} = \lambda X X^T \mathbf{a} \tag{16}$$

When NPE is applied on face image data, the eigenvectors $\mathbf{a}$ are called *NPE-faces*.

Yan et al. [4] discussed that the LLE algorithm can be considered as the direct GE formulation in (3). The matrix M in (14) can be considered as the Laplacian matrix L^{NPE} of the graph G^{NPE}, that is $M = D - W^{NPE} = L^{NPE}$, giving

$$\mathbf{y}^* = \arg\min_{a:\ a^T X X^T a=1} \mathbf{a}^T X L^{NPE} X^T \mathbf{a} \tag{17}$$

This GE formulation of NPE in (17) will help us develop our algorithm.

2.3 Graph View of LDA

LDA is a supervised linear subspace learning algorithm that obtains a discriminant projection function according to class label information of the input data.

The aim of LDA is to find projection directions that maximize the separability of data points belonging to different classes while simultaneously minimizing the distance between data of the same class. Suppose we have N high dimensional image vectors belonging to l classes of faces. LDA maximizes the ratio of the between-class scatter S_b to the within-class scatter S_w, where

$$S_b = \sum_{k=1}^{c} l_k (\overline{\mathbf{x}}^{(k)} - \overline{\mathbf{x}})(\overline{\mathbf{x}}^{(k)} - \overline{\mathbf{x}})^T \tag{18}$$

and

$$S_w = \sum_{k=1}^{c} (\sum_{i=1}^{l_k} (\mathbf{x}_i{}^{(k)} - \overline{\mathbf{x}}^{(k)})(\mathbf{x}_i{}^{(k)} - \overline{\mathbf{x}}^{(k)})^T), \tag{19}$$

where $\overline{\mathbf{x}}$ is the total sample mean vector, l_k is the number of samples in the k-th class and $\mathbf{x}_i^{(k)}$ is the i-th sample in the k-th class. That is, LDA selects optimal $\mathbf{a}$'s as

$$\mathbf{a}^* = \arg\max_a \frac{\mathbf{a}^T S_b \mathbf{a}}{\mathbf{a}^T S_w \mathbf{a}}. \tag{20}$$

Define the total scatter matrix S_t as [8]

$$S_t = \sum_{i=1}^{l} (\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^T. \tag{21}$$

It can be easily shown that

$$S_t = S_b + S_w. \tag{22}$$

Therefore, the optimization (20) can be rewritten as

$$\mathbf{a}^* = \arg\max_a \frac{\mathbf{a}^T S_b \mathbf{a}}{\mathbf{a}^T S_t \mathbf{a}}. \tag{23}$$

That is, the optimal $\mathbf{a}$'s are the eigenvectors corresponding to the largest non-zero eigenvalues of the generalized eigenvalue problem

$$S_b \mathbf{a} = \lambda S_t \mathbf{a} \tag{24}$$

According to Yan et al. [4], the LDA algorithm can also be reformulated as a direct GE by constructing c complete subgraphs $\{G_k\}_{k=1}^c$ each representing data belonging to the corresponding class. Assuming each subgraph G_k has l_k vertices, the weights of each subgraph are defined as an $l_k \times l_k$ weight matrix $W^{(K)}$ with each element equal to $1/l_k$.

Assuming that $X^{(k)} = [X_1^{(k)}, \ldots, X_{l_k}^{(k)}]$ is the data matrix of the k-th class, the between class scatter S_b can be written as

$$S_b = \sum_{k=1}^{c} X^{(k)} W^{(k)} (X^{(k)})^T \tag{25}$$

and the total scatter matrix S_t can be written as

$$S_t = XX^T \tag{26}$$

where X is the data matrix. If the data are ordered based on their class labels, so $X = [X^{(1)}, \ldots, X^{(c)}]$, then the $l \times l$ weight matrix $W_{l\times l}$ of the graph G^{LDA} consisting of all the c subgraphs is defined as

$$W_{l\times l} = \begin{bmatrix} W^{(1)} & 0 & \ldots & 0 \\ 0 & W^{(2)} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & W^{(c)} \end{bmatrix} \tag{27}$$

Then the optimization in (23) can be rewriteen as

$$\mathbf{a}^* = \underset{a}{\operatorname{argmax}} \frac{\mathbf{a}^T X W_{l\times l} X^T \mathbf{a}}{\mathbf{a}^T X X^T \mathbf{a}} \tag{28}$$

This formulation of LDA in (28) was first introduced in [9] and will help us develop our semi-supervised learning method.

3 Semi-supervised Neighborhood Discriminant Embedding

In this section we develop a semi-supervised subspace learning algorithm which incorporates the manifold structure provided by unlabeled data and the discriminant structure provided by labeled data. Cai et al. [5] provided a general framework for semi-supervised subspace learning for Content Based Image Retrieval (CBIR) and discussed the possibility of constructing multiple graphs to learn the intrinsic discriminant structure of the image data. Following the general framework in [5], we construct two graphs; one to encode the neighborhood preserving information based on the NPE method and the other to encode discriminant class label information based on the LDA method. We exploit the information encoded by the two graphs by formulating a constrained optimization problem consisting of the GE objectives of NPE and LDA. The computation of the projection function reduces to solving a general eigenvalue problem. The Semi-supervised Neighborhood Preserving Discriminant Embedding (SNPDE) algorithm enables us to introduce a new image representation and an improved precision for subspace learning and classification of face images.

3.1 The Objective Function

Let $X_l = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_l]$ be the labeled data set and $X_{l+1} = [\mathbf{x}_{l+1}, \ldots, \mathbf{x}_n]$ be the unlabeled data set, where each sample $\mathbf{x}_i$ $(i = 1, \ldots, n)$ is from one of c classes. Let l_k be the number of labeled samples in class k, $(k = 1, \ldots, c)$, so $(\sum_{k=1}^{c} l_k = l)$. Put $X = [\mathbf{X}_l, \mathbf{X}_{l+1}]$ and

$$W^{LDA} = \begin{bmatrix} W_{l\times l} & 0 \\ 0 & 0 \end{bmatrix} \tag{29}$$

where the $W_{l\times l}$ is defined in (27). The SNPDE objective consists of two parts corresponding to the objectives of the graph views of LDA and NPE. Put

$$O_{LDA} = \arg\max_{a} \frac{\mathbf{a}^T X W^{LDA} X^T \mathbf{a}}{\mathbf{a}^T X X^T \mathbf{a}} \tag{30}$$

and

$$O_{NPE} = \arg\min_{a} \mathbf{a}^T X L^{NPE} X^T \mathbf{a}. \tag{31}$$

When sufficient labeled data is not available, the LDA algorithm tends to overfit the objective function. In order to avoid overfitting, we use the regularized version of LDA [3]:

$$\arg\max_{a} \frac{\mathbf{a}^T X W^{LDA} X^T \mathbf{a}}{\mathbf{a}^T X X^T + \alpha J(\mathbf{a})} \tag{32}$$

where $J(a)$ is the regularization term. This term provides us the flexibility to incorporate graph objective of NPE in the GE objective of LDA. The combination of LDA objective with other graph based objective function for subspace learning are discussed and applied in [10] and [5]. We append the graph embedding criteria of NPE as a regularization term to LDA. That is, define

$$O_{SSNPE} = O_{LDA} \text{“}+\text{”} O_{NPE} \tag{33}$$

$$= \arg\max_{a} \frac{\mathbf{a}^T X W^{LDA} X^T \mathbf{a}}{\mathbf{a}^T X X^T \mathbf{a} + \alpha \mathbf{a}^T X L^{NPE} X^T \mathbf{a}} \tag{34}$$

$$= \arg\max_{a} \frac{\mathbf{a}^T X W^{LDA} X^T \mathbf{a}}{\mathbf{a}^T X (\tilde{I} + \alpha L^{NPE}) X^T \mathbf{a}} \tag{35}$$

where $\tilde{I}$ is defined as

$$\tilde{I} = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \tag{36}$$

and I is the $l \times l$ identity matrix.

The objective function in (35) reduces to solving the maximum eigenvalue solution to the generalized eigenvalue problem

$$XW^{LDA}X^T\mathbf{a} = \lambda X(\tilde{I} + \alpha L^{NPE})X^T\mathbf{a} \tag{37}$$

To get a stable solution of (37), the matrix $X(\tilde{I} + \alpha L^{NPE})X^T$ is required to be non-singular [11], which is not the case when the image dimension is larger than the number of image samples. We apply Tikhonov Regularization [12], a well studied solution to ill-posed problems in statistics, to solve the singularity problem. The generalized eigenvalue problem in (37) then becomes

$$XW^{LDA}X^T\mathbf{a} = \lambda \left(X(\tilde{I} + \alpha L^{NPE})X^T + \beta I)\right)\mathbf{a} \tag{38}$$

which has stable solutions for $\beta > 0$.

3.2 The Algorithm

The SNPDE algorithm consists of the following steps.

1. **Construct the labeled graph G^{LDA}:** Construct the $n \times n$ weight matrix W^{LDA} of the labeled graph.
2. **Construct the unlabeled graph G^{NPE}:** Construct the k-nearest neighbor graph matrix W^{NPE} based on (10) and calculate the graph laplacian $L^{NPE} = D - W^{NPE}$, where D is a diagonal matrix with entries the column (since W^{NPE} is symmetric, or row) sums of W^{NPE} that is, $D_{ii} = \sum_j W^{NPE}_{ij}$.
3. **Compute the projection matrix:** The $n \times c$ transformation matrix $A = [\mathbf{a}_1, \ldots, \mathbf{a}_c]$ consists of eigenvectors corresponding to the largest non-zero eigenvalues of the generalized eigenvalue problem in (38). Since W^{LDA} is of rank c, we will have exactly c eigenvectors corresponding to the nonzero eigenvalues.

4. **Embed sample images into $c-$dimensional subspace:** Each image sample can be embedded into $c-$dimensional subspaces by

$$\mathbf{x}_i \rightarrow \mathbf{y}_i = A^T\mathbf{x}_i$$

4 Experiments and Discussions

We present experiments and comparisons to demonstrate the effectiveness of our proposed semi-supervised subspace learning algorithm. In section 4.1 we describe the face image datasets that we used in our experiments. In section 4.2 we illustrate the face representations in lower dimensional subspace. The implementation details and recognition error rates are reported in section 4.3.

4.1 Data Sets

We tested our proposed method on two face databases of CMU PIE [13], and ORL [14]. The CMU PIE database contains 68 subjects with 41,368 images of varying poses, lighting and expressions. The ORL database includes 400 images of 40 individuals under different poses and expressions. In our experiments on the PIE database, we chose the frontal pose $C27$ with varying lighting and illumination which leaves us with 43 images for each subject. In our experiments on ORL database, we used all of the available 400 images in the dataset. Figure 1 and 2 show a sample of images from PIE and ORL databases respectively.

The original images from the CMU PIE database were cropped (The ORL images were already cropped) and the cropped images from both databases were then resized to 32×32 pixels. Each image was represented by a 1024-dimensional vector in the original image space. The training dataset which included labeled and unlabeled data was used to learn a projection matrix to project the high dimensional face images to a lower dimensional subspace. We then applied the nearest neighbor classifier in the subspace to determine the recognition error rate of the unlabeled data and the unseen test data. In all cases the training and the test datasets were randomly selected from the database without mixing between the training and testing data points. The results were averaged over 20 different runs.

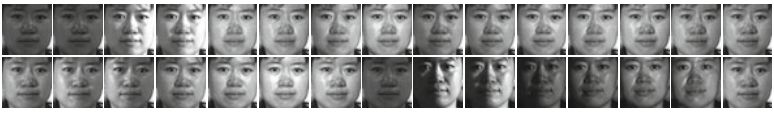
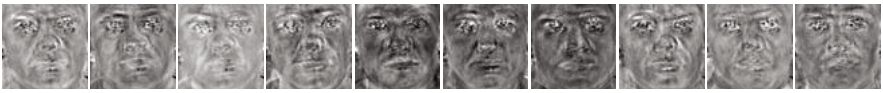


Fig. 1. Sample face images of the CMU PIE face database. Each subject has 43 different images of frontal poses under different lighting conditions.

Fig. 2. Sample face images of ORL face database. Each subject has 10 face images with a different pose and expression.

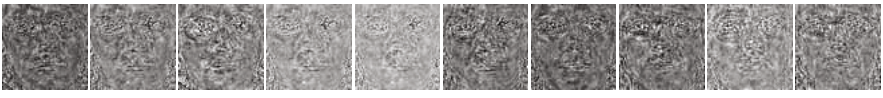
4.2 Face Representation

As mentioned earlier, a high dimensional vector such as the face image vector is prone to the curse of dimensionality and is better studied in lower dimensional subspaces. We compare three algorithms - NPE, SDA, and SNPDE for face representation. In each of these methods, basis functions are thought of as basis images, where each sample image is constructed as a linear combination of the basis images. In Figure. 3, we illustrate first 10 $SNPDEfaces$ together with $NPEfaces$ and $SDAfaces$.



(a) NPEfaces

(b) SDAfaces



(c) SNPDEfaces

Fig. 3. The first 10 NPEfaces, SDAfaces, and SNPDEfaces obtained from samples from the PIE database

4.3 Face Recognition

Table 1 and Table 2 summarize the recognition error rates of four different algorithms. The baseline approach is simply the nearest neighbor approach on the original image space. For the other approaches, training images (labeled and unlabeled) are used to learn a subspace - in NPE approach the training data is constructed in a similar way to the SDA and SNPDE approach, only NPE considers labeled training data as unlabeled. After learning the projection function and projecting the high dimensional data to the image subspace, nearest neighbor classification is performed for recognition purposes. There are two kinds of error rates reported here; the unlabeled error rate, and the test error rate. Although the unlabeled data are used in the training stage, their labels still need to be recognized by the subspace learning algorithm. Therefore, the unlabeled

Table 1. Comparison of recognition error rates on PIE database

Number of Labeled Samples	**Error Rate**(%)							
	Baseline(1024)		**SDA(68)**		**NPE(30)**		**SNPDE(68)**	
	Unlabeled	Test	Unlabeled	Test	Unlabeled	Test	Unlabeled	Test
1	68.10	68.80	61.12	61.23	55.02	57.06	**49.77**	**49.75**
2	56.40	56.49	43.26	43.78	39.93	45.58	**31.45**	**33.61**
3	51.36	46.43	32.74	29.21	36.30	36.19	**23.72**	**22.45**
4	45.13	46.32	25.03	25.26	30.31	33.81	**18.98**	**20.86**
5	39.11	38.47	19.18	18.04	25.55	28.66	**10.80**	**12.70**
6	30.67	33.17	15.49	14.58	20.85	23.37	**8.20**	**8.22**
7	26.89	27.68	12.66	9.64	17.34	20.52	**5.24**	**4.69**
8	29.50	27.25	11.02	9.48	18.85	20.66	**5.92**	**5.88**
9	26.50	24.07	8.81	7.18	16.00	17.88	**4.51**	**4.42**
10	**18.02**	**17.53**	**4.39**	**4.33**	**11.36**	**14.37**	**1.97**	**2.95**

Table 2. Comparison of recognition error rates on ORL database

Number of Labeled Samples	Error Rate(%)							
	Baseline(1024)		SDA(68)		NPE(30)		SNPDE(68)	
	Unlabeled	Test	Unlabeled	Test	Unlabeled	Test	Unlabeled	Test
1	31.03	30.38	27.67	29.38	31.34	39.00	**21.27**	**21.00**
2	17.61	17.38	17.34	22.75	20.32	38.13	**13.80**	**15.13**
3	10.90	11.25	11.50	15.25	11.88	30.88	**7.73**	**10.25**
4	7.10	7.50	9.93	12.50	8.78	30.25	**5.88**	**9.00**
5	**5.34**	**5.25**	**8.00**	**9.00**	**7.47**	**28.50**	**4.69**	**5.75**

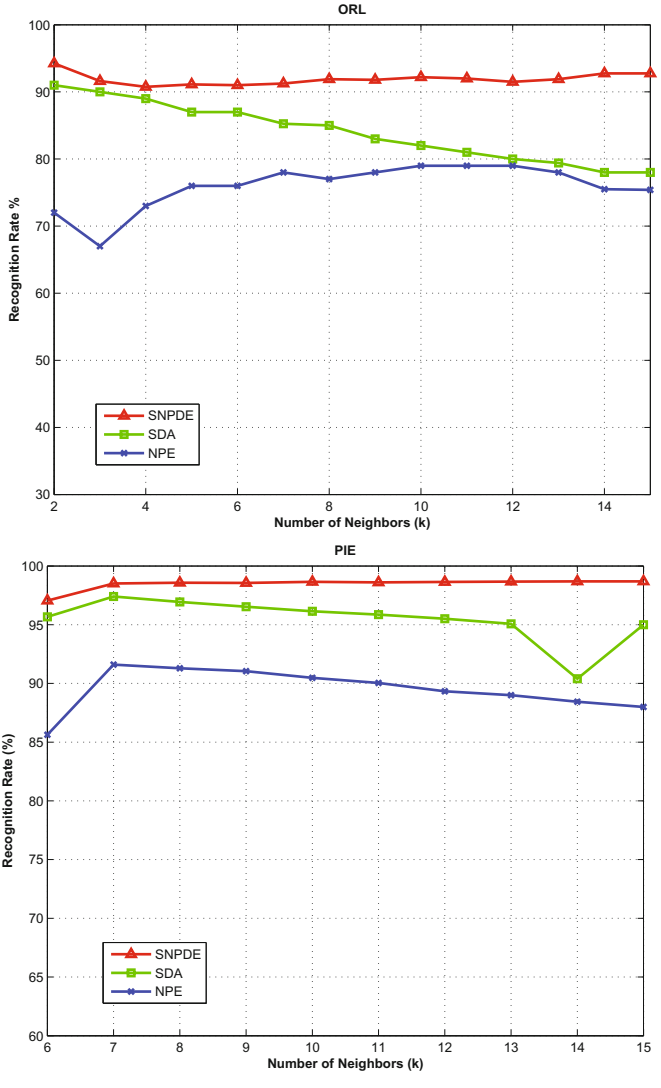


Fig. 4. The effect of the number of neighbors (k) on the performance of the three subspace learning algorithms discussed in this paper

error rate is the error rate associated to the unlabeled data used at the training stage, and the test error rate is the error rate associated with the unseen test images.

The nearest neighbor approach does not consider the manifold structure, and since its decision making is only based on Euclidean distance between images, it provides a very poor performance due to illumination and pose changes. The other approaches learn from the manifold structure, and their difference in performance is due to whether or not they take into account labeled information in their algorithm, and the way the manifold structure is modeled by graphs. SDA is a subspace learning algorithm that considers both labeled and unlabeled data, but since its graph cannot model the manifold structure as accurately as the NPE algorithm does, its performance is inferior to SNPDE.

The error rate of NPE decreases, by increasing the amount of data used in its training stage. The error rate of SDA and SNPDE decreases by increasing the amount of labeled data used at their training stage.

Figure 4 illustrates the sensitivity of three graph-based subspace learning algorithms - NPE, SDA, and SNPDE - to the number of nearest neighbors k in the construction of graphs. The performance of graph-based subspace learning algorithms depend on whether a data point and its nearest neighbors belong to the same class. Therefore, when the number of points in each class in the training data is less than the number of nearest neighbors k, then the possibility of nearest neighbors belonging to different classes increases, consequently reducing the performance of these graph-based methods. This is the case with ORL, a small data set. In contrast, PIE is a large dataset, so in this case all the methods are less sensitive to k. However, the SNPDE still maintains the highest recognition rate of all three algorithms and also is less sensitive to k for both datasets.

5 Conclusion

In this paper, we propose a new linear subspace learning algorithm called Semi-supervised Neighborhood Discriminant Embedding. It can learn from both labeled and unlabeled data to optimize the projection matrix based on both discriminant and geometrical information of high dimensional data. The experimental results on PIE and ORL database demonstrate the effectiveness of our algorithm. As in real applications of biometric face recognition, data becomes available to the system in incremental fashion, we will consider incremental semi-supervised learning based on SNPDE in our future work.

Acknowledgement. We would like to thank Professor Gordon Royle of the Department of Mathematics at the University of Western Australia and Dr. Ashraf Daneshkhah of the Department of Mathematics of the University of Bu-Ali Sina for their valuable discussions and comments. The authors would like to acknowledge the financial support of the Australian Research Council (ARC). This paper is related to ARC DP0771294.

References

1. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. IEEE Trans. Pattern Anal. Mach. Intell. 19, 711–720 (1997)
2. He, X., Cai, D., Yan, S., Zhang, H.: Neighborhood preserving embedding. In: ICCV, pp. 1208–1213 (2005)
3. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer Series in Statistics. Springer, New York (2001)
4. Yan, S., Xu, D., Zhang, B., Zhang, H., Yang, Q., Lin, S.: Graph embedding and extensions: A general framework for dimensionality reduction. IEEE Trans. Pattern Anal. Mach. Intell. 29, 40–51 (2007)
5. Cai, D., He, X., Han, J.: Spectral regression: a unified subspace learning framework for content-based image retrieval. In: ACM Multimedia, pp. 403–412 (2007)
6. Chung, F.R.K.: Spectral Graph Theory. The Regional Conference Series in Mathematics, vol. 92. AMS, Providence (1997)
7. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science 290, 2323–2326 (2000)
8. Fukunnaga, K.: Introduction to Statistical Pattern Recognition, 2nd edn. Academic Press, London (1991)
9. He, X., Yan, S., Hu, Y., Niyogi, P., Zhang, H.: Face recognition using laplacianfaces. IEEE Trans. Pattern Anal. Mach. Intell. 27, 328–340 (2005)
10. Cai, D., He, X., Han, J.: Semi-supervised discriminant analysis. In: ICCV, pp. 1–7 (2007)
11. Watkins, D.S.: Fundamentals of matrix computations. John Wiley & Sons, Inc., New York (1991)
12. Lauter, H., Liero, H.: Ill-posed inverse problems and their optimal regularization (1997)
13. Sim, T., Baker, S., Bsat, M.: The CMU Pose, Illumination, and Expression (PIE) Database of Human Faces. Technical Report CMU-RI-TR-01-02, Robotics Institute, Pittsburgh, PA (2001)
14. ORL database: Cambridge University Computer Laboratory (2002), http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

Segmentation via NCuts and Lossy Minimum Description Length: A Unified Approach

Mingyang Jiang, Chunxiao Li, Jufu Feng, and Liwei Wang

Key Laboratory of Machine Perception (Peking University),
MOE, Department of Machine Intelligence,
School of Electronics Engineering and Computer Science,
Peking University, Beijing 100871, P.R. China
{jiangmy,licx,fjf,wanglw}@cis.pku.edu.cn

Abstract. We investigate a fundamental problem in computer vision: unsupervised image segmentation. During the last decade, the Normalized Cuts has become very popular for image segmentation. NCuts guarantees a globally optimal solution in the continuous solution space, however, how to automatically select the number of segments for a given image is left as an open problem. Recently, the lossy minimum description length (LMDL) criterion has been proposed for segmentation of images. This criterion can adaptively determine the number of segments, however, as the optimization is combinatorial, only a suboptimal solution can be achieved by a greedy algorithm. The complementarity of both criteria motivates us to combine NCuts and LMDL into a unified fashion, to achieve a better segmentation: given the NCuts segmentations under different numbers of segments, we choose the optimal segmentation to be the one that minimizes the overall coding length, subject to a given distortion. We then develop a new way to use the coding length decrement as the similarity measure for NCuts, so that our algorithm is able to seek both the optimal NCuts solution under fixed number of segments, and the optimal LMDL solution among different numbers of segments. Extensive experiments demonstrate the effectiveness of our algorithm.

1 Introduction

A fundamental problem in computer vision is to automatically partition a natural image into regions with homogeneous texture, commonly refers to as image segmentation. Segmentation is widely accepted as a crucial function for many visual tasks such as object recognition, scene understanding and monocular inference of 3D structure. These recent vision applications have led to a renewed interest in automatic image segmentation algorithms.

In the literature, investigators have explored several important models and criteria that can lead to good image segmentation. Traditional clustering algorithms aim at extracting the statistical characteristics of the region data, such as k-means [1] and Mean Shift [2]. The graph based region merging algorithm F&H [3] attempts to partition image into regions such that the resulting segmentation is neither too coarse nor too fine. While region contours/edges contain

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 213–224, 2011.

useful shape information about the saliency of the objects in the image [4], several approaches have been proposed to combine the cues of homogeneous color or texture with contours in the segmentation process [5] [6].

In recent years, much attention has been paid to spectral clustering algorithms [7], in particular the Normalized Cuts criterion [8], which provides a way of integrating global image information into the grouping process. The original NCuts criterion is concerned on the 2-way situation, which aims at partitioning image into two parts. Two recent variants extent the NCuts criterion to the k-way multi-class and multi-scale situation, known as the Multi-class NCuts [9] and Multi-scale NCuts [10]. These progress address segmentation in a k-way global optimization framework and guarantee a globally optimal solution in the relaxed continuous solution space. However, the k-way NCuts criterion can not automatically select the number of segments, k, since the objective function of k-way NCuts increases monotonically as k is varied. In many applications such as natural image segmentation, due to the diversity and complexity of image contents and semantemes, the optimal number of segments k may be different for varying images. For unsupervised segmentation, how to adaptively select the number of segments k for varying images is a fundamental open problem left in [9] and [10]. We also note that how to construct affinity matrix is another important issue in NCuts framework, which significantly influences the segmentation performance [5] [11].

More recently, an objective criterion based on the notion of lossy minimum description length (LMDL) has been proposed for evaluating segmentation of images [12]. The "optimal segmentation" is defined as the one that minimizes the number of bits needed to code the segmented data, subject to a given distortion. The most recent progress based on LMDL [13] [14] [15] have shown that this criterion is highly consistent with human segmentation of images. Preliminary success of LMDL suggests that: firstly, it is appropriate for evaluating segmentation performance objectively; secondly, the coding length serves as a reliable similarity measure between pairs of regions; last, but not the least, LMDL can adaptively determine the optimal number of segments for a given image. However, as the minimization problem is NP hard, a suboptimal solution is found by iteratively merging regions to reduce the coding length. There is no theoretical proof for the optimality of the greedy algorithm.

Although there are numerous criteria that address segmentation problem, there is little consensus on what criteria strike a best balance between objective measures that depend solely on the intrinsic statistics of imagery data and subjective measures that try to empirically mimic human perception. Some recent works such as [16] [17] focus on giving a unified perspective and evaluation procedure addressing the problem "what is a good segmentation".

Paper contributions. In this paper, we contend that, much better results can be obtained by properly combining different criteria for segmentation into a unified fashion. We propose a unified framework combining both NCuts and

LMDL criteria, to achieve an "optimal segmentation". The main novelty and the specific contributions of this paper are as follows:

1. We propose a method to automatically select the number of segments for Multi-class NCuts, using LMDL criterion. The optimal number of segments in NCuts is the one that the corresponding segmentation minimizes the overall coding length, subject to a given distortion. This perspective combines both NCuts and LMDL criteria into a unified framework, to achieve the optimal segmentation of given data.
2. We develop a new way to use the coding length decrement directly as a pairwise affinity measure, to build the affinity matrix in NCuts. This procedure allows the proposed algorithm to seek both the optimal NCuts solution under fixed number of segments and the optimal LMDL solution among different numbers of segments, thus achieves a better segmentation.
3. The experiments validate that the proposed algorithm captures the advantages of both NCuts and LMDL, thus achieves comparable or even better segmentation results compared with the state-of-the-arts.

2 Related Work

We begin by reviewing Multi-class NCuts criterion and lossy minimum description length criterion, which are closely related to our work.

2.1 k-Way Normalized Cuts Criterion

Here, we focus on the k-way Normalized Cuts [9], which means partitioning an image into k segments. Given an image I, we construct a graph $G = (V, E, W)$. Here the graph nodes V can represent either pixels or "superpixels", which is a commonly used initiation in image segmentation [18]. Suppose there are in total N nodes in the graph. Each pair of nodes is connected by a graph edge E. A weight value $W(i, j)$ represents the affinity between nodes i, j, which measures the likelihood of nodes i and j belonging to the same image segments. For a bipartition of the graph $V = V_1 \cup V_2 \cup \cdots \cup V_k, \forall V_i \cap V_j = \phi, i \neq j$, the k-way Normalized Cuts criterion is defined as:

$$\min kNcuts(V) = \frac{1}{k} \sum_{l=1}^{k} \frac{cut(V_l, V \backslash V_l)}{assoc(V_l, V)} \tag{1}$$

In the above equation, $cut(V_l, V \backslash V_l) = \sum_{i \in V_l, j \in V \backslash V_l} W(i, j)$ measures how many edge weights escape from V_l. $assoc(V_l, V) = \sum_{i \in V_l, j \in V} W(i, j)$ measures how many edge weights connects V_l.

Although directly optimizing the k-way NCuts is NP-hard, relaxing the partition indication matrix into the continuous domain turns it into a tractable

continuous optimization problem and can be solved by eigenvalue decomposition of the normalized affinity matrix. This procedure is commonly known as spectral relaxing [9]. Based on the relaxed continuous solution, the final discrete solution is obtained by spectral rotation.

From Corollary 1 in [9], the k-way NCuts objective increases monotonically as k increases. This result indicates that k-way NCuts can not adaptively select the number of segments k for a given image. Consequently, *how to adaptively choose k remains an open problem.*

2.2 Lossy Minimum Description Length Criterion

In [12], Ma et.al proposed an objective measure to evaluate the quality of segmentations, which is based on the lossy minimum description length (LMDL) criterion. This criterion draws strong connection between data compression and segmentation. The optimal segmentation is defined to be the one that minimizes the number of bits needed to code the segmented data, subject to a given distortion.

First, we consider a single region V_i with m_i pixels. Based on [12], for a fixed distortion ε, the number of bits needed to code V_i under Gaussian case can be written as:

$$L(V_i) = \frac{m_i + p}{2} \log_2 \det(I + \frac{p}{\varepsilon^2} \Sigma_i) + \frac{p}{2} \log_2(1 + \frac{\mu_i^T \mu_i}{\varepsilon^2}), \tag{2}$$

where μ_i and Σ_i are the sample mean and variance of region V_i, p is the sample dimension of data.

Suppose an image I can be segmented into non-overlapping regions $V = V_1 \cup V_2 \cup \cdots \cup V_k, \forall V_i \cap V_j = \phi, i \neq j$. The LMDL criterion seeks to minimize the overall coding length of the image I:

$$min \;\; L(V_1, V_2, ..., V_k) = \sum_{i=1}^{k} [L(V_i) + m_i(-\log_2(m_i/m))] \tag{3}$$

In the above expression, m is the total number of pixels in an image, i.e., $m = m_1 + m_2 + ... + m_k$. The second term is the number of bits needed to code the membership of the m samples in the k groups (using the Huffman coding).

It is worth noticing that, once the distortion ε is fixed, the number of segments in the segmentation is *automatically determined* [12]. This completely avoids the necessity of additional interaction usually required with traditional segmentation methods, such as k-way NCuts.

However, as this minimization problem is combinatorial, all the LMDL based algorithms seek a suboptimal solution via an agglomerative way: first initialize superpixels and assume each superpixel forms its own group, then iteratively merge adjacent pair of regions that yields *the largest decrease in coding length* until the overall coding length achieves a local minima.

3 LMDL-NCuts: A Combined and Unified Criterion

We now describe our approach, which aims at combining both NCuts and LMDL into a unified fashion. We describe the general criterion below, then discuss the construction of the affinity matrix using the coding length decrement.

3.1 The Combined Criterion

From the previous section, it is clear that k-way NCuts addresses segmentation in a global optimization framework and guarantee a globally optimal solution in the continuous solution space. However, how to adaptively choose the number of segments k is left as an open problem, especially in the semantically complicated scenario such as natural image segmentation. On the other side, LMDL criterion can automatically determine the number of segments. However, the LMDL optimization is combinatorial, only local minima can be guaranteed. The complementarity of both criteria motivates us to combine both criteria into a unified fashion, to achieve a better segmentation. That is, *given the NCuts segmentations under different ks, we choose the optimal segmentation to be the one that minimizes the overall coding length, subject to a given distortion.* We refer this combined criterion to as LMDL-NCuts criterion. Note that under this criterion, the optimal number of segments is adaptively determined.

3.2 Initializing

In the original NCuts algorithm, the segmentation is directly performed on the image pixels. There are two problems for such a processing. First, each pixel will be a node in the graph so that the computational cost will be very high. Second, two pixels are connected in a graph if and only if their spatial distance is smaller than a graph connection radius G_r [10], which makes the original NCuts not able to catch the global graph topology and information. It has been investigated in [10] that larger G_r generally makes segmentation better. These problems can be solved by initializing an image with millions of pixels into a few hundred or thousand "superpixels". A superpixel is a small region that does not contain strong edges in its interior. There are several algorithms that can be used to obtain a superpixel initialization [3] [19] [20]. We have compared the three methods in the experiments and found that [20] works best for our purposes[1]. Such superpixel initialization greatly reduce the computational cost, and also, the graph connection radius constraint is no longer a necessity in our approach.

3.3 Construct Affinity Matrix

Since we build the segmentation based on the superpixel level, how to define the similarity between two superpixels is another important issue.

[1] We use the publicly available code for this method available at `http://www.cs.sfu.ca/~mori/research/superpixels/` with parameter $N_{sp} = 200$.

Since we are potentially seeking for the segmentation that can yield minimum lossy coding length, a direct way is to link the coding length with the similarity measure. In the LMDL based algorithms, the coding length decrement is used implicitly as a similarity measure between pair of regions, i.e., the decrease in the coding length essentially captures the similarity of the regions [15][2]. The larger coding length decrement means the pair of regions is more similar, so that they are merged in the LMDL based algorithms. Previous success of the LMDL based algorithms [12] [13] [15] suggests that the coding length decrement is a reliable similarity measure between regions. In our work, we directly use the coding length decrement as the affinity $W(i,j)$ between superpixels V_i and V_j. Based on Equation 2 and 3, the affinity can be written as:

$$W(i,j) = L(V_i, V_j) - L(V_i \cup V_j)$$

$$= \log_2 [\frac{\left|I+\frac{p}{\varepsilon^2}\sum_i\right|^{\frac{m_i+p}{2}}\left|I+\frac{p}{\varepsilon^2}\sum_j\right|^{\frac{m_j+p}{2}}}{\left|I+\frac{p}{\varepsilon^2}\sum\right|^{\frac{m_i+m_j+p}{2}}} \cdot \frac{(1+\frac{\mu_i^T\mu_i}{\varepsilon^2})^{\frac{p}{2}}(1+\frac{\mu_j^T\mu_j}{\varepsilon^2})^{\frac{p}{2}}}{(1+\frac{\mu^T\mu}{\varepsilon^2})^{\frac{p}{2}}} \cdot (\frac{m}{m_i})^{m_i}(\frac{m}{m_j})^{m_j}]$$

Here (μ, Σ) are the sample mean and variance of region $V_i \cup V_j$, respectively. Note that $W(i,j) < 0$ means merging V_i, V_j will increase the overall coding length, which indicates that V_i, V_j are dissimilar. In this case, we simply set $W(i,j) = 0$. The calculated affinities are then normalized into the range [0,1]. The NCuts optimization is conducted based on the affinity matrix.

3.4 The Algorithm

Based on the previous discussion, we summarize the overall segmentation algorithm in Algorithm 1, which we refer to as *LMDL-NCuts Segmentation* (LNC).

Algorithm 1. LNC

1. **Input**: image data I, distortion ε;
2. **Initialization:** superpixels as graph nodes: $V := \{V_i = \{v\} \,|v \in V\}$;
3. Construct affinity matrix based on the coding length decrement;
4. **for** $k = 1 : M$ **do**
5. SegmentationResults(k) = NCuts(k);
6. Choose SegmentationResults(i) that yields the smallest overall coding length;
7. **Output**: SegmentationResults(i).

Note that for the maximum number of segments M, one can choose it to be the initial number of superpixels. However, we found that for most natural images, the number of segments larger than 40 leads to serious oversegmentation. So in our experiments, we set $M = 40$.

[2] The general spirit of a bottom-up segmentation process is to merge the "similar" regions. In the LMDL based algorithms, the pair of regions that yields the largest decrease in the coding length is merged in each iteration, which means they are "most similar" measured by the coding length similarity measure.

4 Experiments

In this section, we conduct extensive evaluation on two publicly available datasets: Berkeley Segmentation Dataset [21] and MSRC Object Recognition Dataset [22], to validate the performance of the proposed LNC algorithm. We will first describe the features used in our method. In section 4.2, we will validate that LMDL-NCuts is effective on selecting the appropriate number of segments. In section 4.3, we will discuss the effect of distortion parameter and make a close comparison with LMDL based algorithm. Both qualitative and quantitative results compared with the state-of-the-arts are listed in section 4.4.

4.1 Feature Construction

As shown in Figure 1, given an image, we convert it to the $L*a*b$ color space. In order to capture the variation of a local texture, we directly use the 7×7 cut-off window around each pixel and stack the color values inside the window into a vector form. Each texture window is smoothed by convolving with a 2D Gaussian kernel before stacking. Finally, for the ease of computation, we project the feature vectors into a D-dimensional space using PCA. We have observed that for most natural images, the first eight principal components of original feature data contain over 99% of the energy. So we set $D = 8$. The feature extraction and pre-processing are similar as in [15].

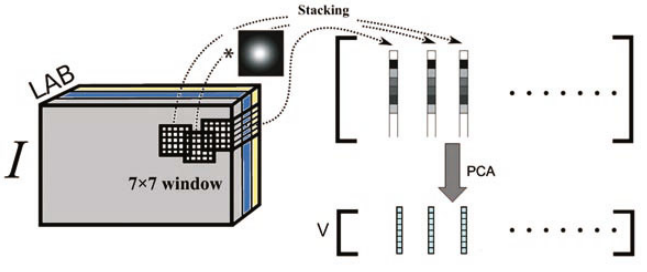


Fig. 1. Feature construction. The 7×7 windows around each pixel on the $L*a*b$ color space are convoluted with 2D Gaussian kernel, stacked into a one column vector and then use PCA.

4.2 Adaptively Select the Number of Segments

We conduct experiments on the MSRC Object Recognition Dataset [22] to validate that our LNC algorithm can adaptively select the appropriate number of segments for NCuts. MSRC dataset consists of 591 images grouped into 20 categories. Each image consists of a salient object, and the background is not so complicated so that it is visually not hard to validate the appropriate number of segments. We found that under $\varepsilon = 0.20$, the minimum description length solution provides the best visually appealing results, i.e., for most of the images, our algorithm can adaptively determine a reasonable number of segments. Some sample results are listed in Figure 2.

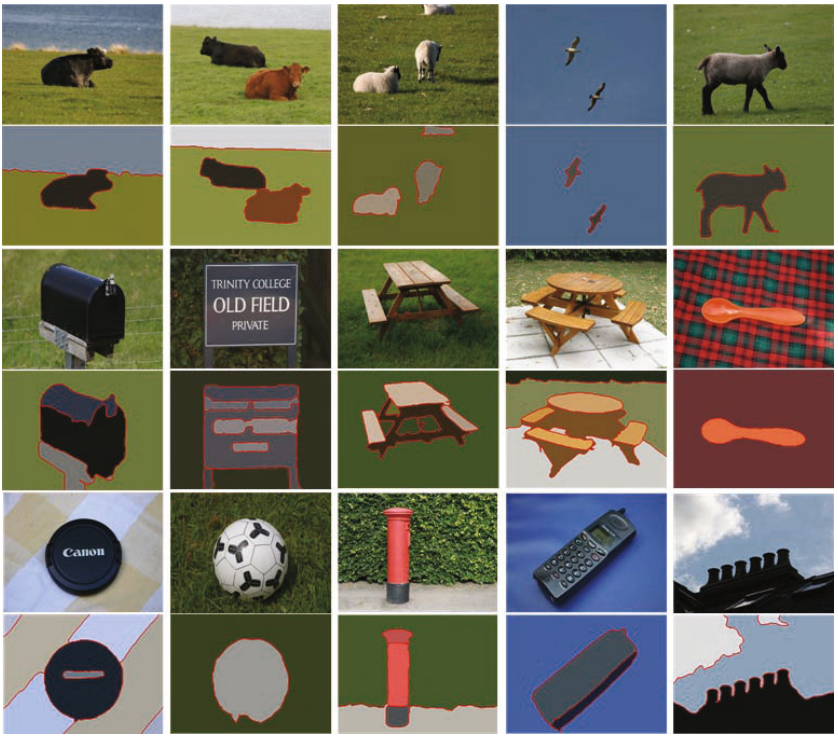


Fig. 2. (color) Qualitative results on the MSRC Object Recognition Database. For each result, the top is the original image, and the bottom is the segmentation results with each region colored by its mean color.

4.3 Comparison with LMDL Based Algorithm

We then conduct experiments on the Berkeley Segmentation Dataset (BSDS) [21]. BSDS consists of 200 training and 100 test images, and each of them has been manually segmented by a number of different subjects. Since the proposed algorithm is based on the LMDL criterion, it is worth comparing our algorithm with the LMDL based algorithms. Because we use the coding scheme proposed in the pioneer work [12], in this section, we compare LNC with the algorithm proposed in [12], namely Pairwise Steepest Descent (PSD). More comparison with other LMDL based algorithms can be found in section 4.4.

Note that the distortion ε is the only parameter in LMDL criterion. In the experiment, we compare PSD's results with our results under 4 different choices of ε: 0.10, 0.15, 0.20 and 0.25. We use three quantitative measures to evaluate the segmentation results: the average overall coding length (under given distortion), the Probability Rand Index (PRI) [23] and the Variation of Information (VoI) [24]. The objective of LMDL is to seek for the smallest coding length, so the smaller average overall coding length, the better the segmentation is. PRI and VoI aim at comparing the segmentation results with ground-truth

Table 1. Comparison between LNC algorithm and PSD algorithm. For average coding length, lower is better. For PRI, higher is better. For VoI, lower is better. Boldface indicates the better results.

	$\varepsilon = 0.10$		$\varepsilon = 0.15$		$\varepsilon = 0.20$		$\varepsilon = 0.25$	
Method/Index	LNC	PSD	LNC	PSD	LNC	PSD	LNC	PSD
avg. code length (kb)	2361	**2321**	1868	**1865**	**1562**	1573	**1331**	1364
PRI	**0.777**	0.756	**0.783**	0.758	**0.791**	0.748	**0.776**	0.724
VoI	**2.069**	2.316	**1.883**	2.062	**1.804**	1.925	**1.763**	1.896

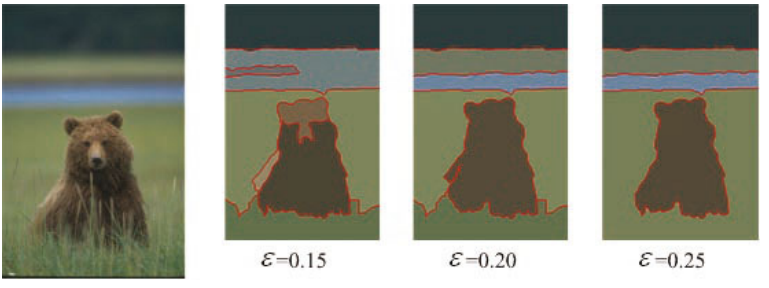


Fig. 3. (color) Segmentation results under different ε. Left: Input image. Right: Segmentation results under $\varepsilon = 0.15, 0.20, 0.25$, respectively.

results. For PRI index, the larger, the better. And for VoI, the smaller, the better. For brevity, we refer the reader to the stated references for the definition of each metric.

The results are listed in Table 1. For both PRI and VoI under all choices of ε, LNC consistently outperforms PSD. As choosing the distortion is the main difficulty in the LMDL based algorithms, here we note that the LNC algorithm is less sensitive to the choice of distortion. And, as illustrated in Figure 3, the effect of the distortion to the segmentation result is the same as in [15]: smaller choice of ε turns to over-segment images and larger ε turns to under-segment images. We also note an interesting result that LNC achieves *comparable or even smaller* average coding length compared with PSD, which is designed to directly minimize the coding length. These results suggest that the LNC algorithm captures both advantages of LMDL and NCuts, thus achieves a better segmentation.

4.4 Qualitative and Quantitative Comparison

We compare the performance of the LNC algorithm with five *publicly available* image segmentation methods: Mean-Shift (MS) [2], F&H [3], Multi-scale NCuts (MNCuts) [10], Compression-based Texture Merging (CTM) [13] and Texture and Boundary Encoding-based Segmentation (TBES) [15], on the Berkeley

Table 2. Quantitative comparison on the BSDS. Boldface indicates the best results.

Index/Method	Human	LNC	MS	FH	MNCuts	CTM	TBES
PRI (Higher is better)	0.868	**0.791**	0.772	0.770	0.742	0.742	0.787
VoI (Lower is better)	1.163	**1.804**	2.203	2.844	2.651	2.002	1.824

Fig. 4. (color) Qualitative results of LNC algorithm on the BSDS. For each result, the top is the original image, and the bottom is the segmentation with each region colored by its mean color.

Segmentation Dataset (BSDS). The performance of these five methods and that of human's, based on PRI and VOI measures, were obtained by personal communication with the authors of [15]. The user-defined parameters of these methods

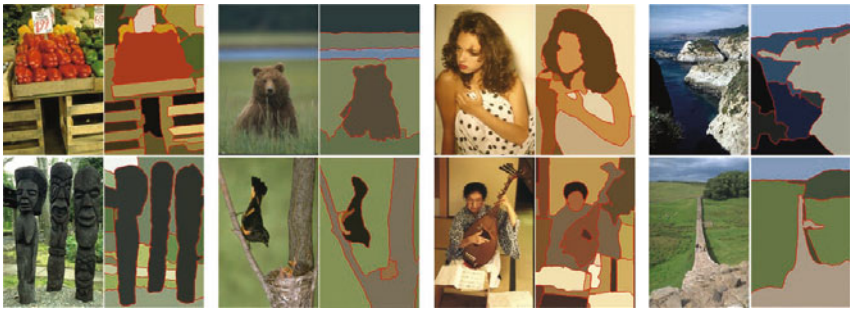


Fig. 5. (color) Qualitative results of LNC algorithm on the BSDS

have been tuned to achieve the best overall tradeoff between PRI and VoI. In particular, we report our results under $\varepsilon = 0.20$. Table 2 summarize the quantitative results on the BSDS.

Among all the algorithms in Table 2, LNC achieves the best result on both PRI and VoI. More qualitative results are shown in Figure 4 and 5.

5 Conclusion

In this paper, we have proposed a unified approach to image segmentation. It seeks both the optimal NCuts solution under fixed number of segments, and the optimal LMDL solution among segmentations under different numbers of segments. Our approach can automatically select the number of segments for NCuts, subject to a given distortion in LMDL criterion.We also develop a novel way to directly use the lossy coding length decrement as the affinity measure between superpixels, and use this affinity measure to construct affinity matrix. The experiments validate that the proposed algorithm can adaptively select the appropriate number of segments for a given image, and can yield comparable or even smaller overall coding length compared with the LMDL based algorithms. The segmentation results match well with human segmentations, compete or exceed with the best segmentation algorithms.

Acknowledgement. This work was supported by NBRPC(2011CB302400), NSFC(60635030), NSFC(61075003) and NSFC(60775005).

References

1. Duda, R., Hart, P., Stork, D.: Pattern Classification. John Wiley and Sons, Chichester (2001) 0-471-05669-3
2. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. PAMI 24, 603–619 (2002)
3. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. IJCV 59, 167–181 (2004)
4. Ren, X., Fowlkes, C., Malik, J.: Learning probabilistic models for contour completion in natural images. IJCV 77, 47–63 (2008)

5. Malik, J., Belongie, S., Leung, T.K., Shi, J.: Contour and texture analysis for image segmentation. IJCV 43, 7–27 (2001)
6. Zhu, S.C., Tu, Z.W.: Image segmentation by data-driven markov chain monte carlo. PAMI II, 131–138 (2002)
7. Chung, F.R.K.: Spectral graph theory. Regional Conference Series in Mathematics, vol. 92, pp. 1–212. American Mathematical Society, Providence (1997)
8. Shi, J., Malik, J.: Normalized cuts and image segmentation. PAMI 22, 888–905 (2000)
9. Yu, S.X., Shi, J.B.: Multiclass spectral clustering. In: ICCV, pp. 313–319 (2003)
10. Cour, T., Benezit, F., Shi, J.B.: Spectral segmentation with multiscale graph decomposition. In: CVPR, vol. II, pp. 1124–1131 (2005)
11. Kim, T.H., Lee, K.M., Lee, S.U.: Learning full pairwise affinities for spectral segmentation. In: CVPR (2010)
12. Ma, Y., Derksen, H., Hong, W., Wright, J.: Segmentation of multivariate mixed data via lossy data coding and compression. PAMI 29, 1546–1562 (2007)
13. Yang, A.Y., Wright, J., Ma, Y., Sastry, S.S.: Unsupervised segmentation of natural images via lossy data compression. CVIU 110, 212–225 (2008)
14. Rao, S.R., Mobahi, H., Yang, A.Y., Sastry, S.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)
15. Mobahi, H., Rao, S.R., Yang, A.Y., Sastry, S.S., Ma, Y.: Segmentation of natural images by texture and boundary compression. In: arXiv:1006.3679v1 (2010)
16. Bagon, S., Boiman, O., Irani, M.: What is a good image segment? A unified approach to segment extraction. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 30–44. Springer, Heidelberg (2008)
17. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: From contours to regions: An empirical evaluation. In: CVPR, pp. 2294–2301 (2009)
18. Ren, X.F., Malik, J.: Learning a classification model for segmentation. In: ICCV, pp. 10–17 (2003)
19. Ren, X.F., Fowlkes, C.C., Malik, J.: Scale-invariant contour completion using conditional random fields. In: ICCV, vol. II, pp. 1214–1221 (2005)
20. Mori, G.: Guiding model search using segmentation. In: ICCV (2005)
21. 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: ICCV, vol. 2, pp. 416–423 (2001)
22. Shotton, J., Winn, J., Rother, C., Criminisi, A.: Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 1–15. Springer, Heidelberg (2006)
23. Rand, W.M.: Objective criteria for the evaluation of clustering methods. American Statistical Association Journal 66, 846–850 (1971)
24. Meila, M.: Comparing clusterings: An axiomatic view. In: ICML (2005)

A Phase Discrepancy Analysis of Object Motion

Bolei Zhou[1,2,*], Xiaodi Hou[3,*], and Liqing Zhang[1]

[1] MOE-Microsoft Key Laboratory for Intelligent Computing and Intelligent Systems, Dept. of Computer Science and Engineering, Shanghai Jiao Tong University
[2] Dept. of Information Engineering, The Chinese University of Hong Kong
[3] Dept. of Computation and Neural Systems, California Institute of Technology
zhoubolei@gmail.com, xiaodi.hou@gmail.com, zhang-lq@cs.sjtu.edu.cn

Abstract. Detecting moving objects against dynamic backgrounds remains a challenge in computer vision and robotics. This paper presents a surprisingly simple algorithm to detect objects in such conditions. Based on theoretic analysis, we show that 1) the displacement of the foreground and the background can be represented by the phase change of Fourier spectra, and 2) the motion of background objects can be extracted by *Phase Discrepancy* in an efficient and robust way. The algorithm does not rely on prior training on particular features or categories of an image and can be implemented in 9 lines of MATLAB code.

In addition to the algorithm, we provide a new database for moving object detection with 20 video clips, 11 subjects and 4785 bounding boxes to be used as a public benchmark for algorithm evaluation.

1 Introduction

Detecting moving objects in a complex scene is one of the most challenging problems in computer vision. It is closely related to a variety of critical applications such as tracking, video analysis, content retrieval, and robotics. Generally speaking, motion detection methods can be categorized into three main approaches: background modeling, detection by recognition, and view geometry.

Many models try to attack the problem of detection under controlled situations. For instance, some algorithms assume a stationary camera. This assumption leads to a branch of techniques called background subtraction. The main idea is to learn the appearance model of the background [1] [2]. A moving object in the scene is then detected by subtracting the background image from the current image. However, scene appearance captured by a moving camera, with foreground and backgrounds in arbitrary depths and viewpoints, can be very complicated. Thus, most of the background models perform poorly on moving camera recordings [3].

Another branch of popular algorithms stems from object detection and recognition. Based on pre-trained detectors, an algorithm can detect objects from particular categories, such as faces [4] or pedestrians [5]. These algorithms usually require offline training and can only handle a very limited number of object categories. Moreover, finding an invariant object detector that overcomes

[*] These two authors contributed equally to this paper.

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 225–238, 2011.

Fig. 1. An illustration of moving object detection from a perspective of optical flow analysis. **A**): A video sequence with both object motions and camera motion. **B**): The corresponding optical flow. **C**): The segmentation result that detects the moving objects.

illumination/view-point changes and occlusion, is already a challenge in computer vision.

To circumvent these problems, some other algorithms detect motion via camera geometry [6] [7]. This approach estimates the camera parameters under certain geometric constraints, use these parameters to compensate for camera-induced motion, and separate the moving object from the residual motion in the scene [8].

In principle, a visual system needs *only* motion cues to detect an moving object – even if the scene is disturbed by camera's ego-motion. With full knowledge of the optical flow, the mission of object detection is to find the cluster of consistent motion that is induced by the foreground. Nevertheless, the computational burdens of an optical flow algorithm is usually very heavy.

Related Work. In 2001, Vernon [9] proposed using a Fourier transform to untangle the complexity of object motions. In his theory, object segmentation and exact velocity recovery can be achieved by solving a linear system. Based on the translation property of Fourier transform, a moving object corresponds to a phase change in the Fourier spectrum. For a scene composed of m objects, exact recovery is achieved by solving a linear equation with $2m$ unknowns. The drawback of this approach is that the number m of objects must be specified beforehand. Moreover, the segmentation and velocity recovery requires observing $2m$ frames, which every object moving at a constant speed. These constraints preclude Vernon's approach from real-world applications.

An Outline of Our Approach. We start from a similar perspective to that of Vernon: spatially distributed information can be efficiently accumulated in the Fourier spectrum. However, instead of finding the exact solution for a constrained problem, we find an approximate solution using a minimal number of assumptions.

To extract moving objects from dynamic backgrounds, our model follows the idea of predictive coding. First, we predict the next frame only considering background movements. Then by comparing our prediction against the actual observation, pixels representing the foreground emerge due to the large reconstruction error. With rigorous analysis, we show that a 9-line MATLAB approximation recovers the camera motion with bounded error.

2 The Theory

We denote $f(\mathbf{x}, t)$ as our observation at time t[1], where $\mathbf{x} = [x_1, x_2]^\top$ is the 2-dimensional vector of a spatial location. Let $\mathcal{I}$ be the ensemble of pixels. For any image, we have the partition $\mathcal{I} = \{\mathcal{F}_t, \mathcal{B}_t\}$. Every pixel belongs to the foreground $\mathcal{F}_t$ or the background $\mathcal{B}_t$.

For typical sampling rates, the ego-motion of the camera is well approximated by a uniform translation of the background. If we know this displacement $\mathbf{v} = [v_1, v_2]^\top$, we can predict the appearance of the background in the next frame based on the *intensity constancy* assumption [10] that the spatial translation does not change pixel values:

$$f(\mathbf{x}, t) = f(\mathbf{x} + \mathbf{v}, t + 1), \quad \text{where } \mathbf{x} \in \mathcal{B}_t \bigcap \mathcal{B}_{t+1} \tag{1}$$

This assumption requires that pixels $\mathbf{x}$ at t and $\mathbf{x} + \mathbf{v}$ at $t + 1$ belong to the background. We further denote $\check{\mathcal{B}}_t = \hat{\mathcal{B}}_{t+1} = \mathcal{B}_t \bigcap \mathcal{B}_{t+1}$.

Once we have the ground-truth of the ego-motion, we can reconstruct the next frame by shifting every pixel from $\mathbf{x}$ to $\mathbf{x} + \mathbf{v}$. This reconstruction is expected to perform poorly for pixels in $\mathcal{I} - \check{\mathcal{B}}_t$, the foreground. Thus, we can take the error as a likelihood function of the appearance of moving objects at certain locations. In other words, the reconstruction error map $s(\mathbf{x}, t)$ can be considered as a *saliency map* [11] for moving objects:

$$s(\mathbf{x}, t) = \Big[f(\mathbf{x} + \mathbf{v}, t + 1) - f(\mathbf{x}, t) \Big]^2. \tag{2}$$

2.1 Phase Discrepancy and Ego-motion

In order to generate the saliency map, we need to know the displacement vector $\mathbf{v}$. In the Fourier domain, the spatial displacement in Eq.1 can be efficiently represented by the phase of the Fourier spectrum.

Let $F_{\mathbf{x}_i,t}(\boldsymbol{\omega}) = \mathcal{F}[f(\mathbf{x}, t) \cdot \delta_{\mathbf{x}_i}(\mathbf{x})]$ denote the 2-D Discrete Fourier transform of a single pixel, where $\boldsymbol{\omega} = [\omega_1, \omega_2]^\top$, and the indicator function $\delta_{\mathbf{x}_i}(\mathbf{x})$ is defined as:

$$\delta_{\mathbf{x}_i}(\mathbf{x}) = \begin{cases} 1 \text{ if } \mathbf{x} \in \mathbf{x}_i, \\ 0 \text{ otherwise.} \end{cases}$$

The Fourier spectrum of the entire image $F_t(\boldsymbol{\omega})$ can be obtained by:

$$F_t(\boldsymbol{\omega}) = \sum_{\mathbf{x}_i \in \mathcal{I}} F_{\mathbf{x}_i,t}(\boldsymbol{\omega})$$

Known as the translation property [12], a spatial displacement entails a phase change, yet leaves the Fourier amplitudes intact:

$$F_{\mathbf{x}+\mathbf{v},t+1}(\boldsymbol{\omega}) = F_{\mathbf{x},t}(\boldsymbol{\omega}) e^{-i \cdot \Phi(\mathbf{v})}, \tag{3}$$

[1] For simplicity, we only consider gray-scale images in this section. A simple extension to color images is provided in Section 3.

where $\Phi(\mathbf{v}) = \boldsymbol{\omega}^\top \mathbf{v} = \omega_1 v_1 + \omega_2 v_2$, which we call the *phase discrepancy* in the following discussions.

Because the entire background has approximately the same displacement $\mathbf{v}$, Eq.3 has a compact form for $\check{\mathcal{B}}_t$:

$$\sum_{\mathbf{x}_i \in \hat{\mathcal{B}}_{t+1}} F_{\mathbf{x}_i,t+1}(\boldsymbol{\omega}) = \sum_{\mathbf{x}_i \in \check{\mathcal{B}}_t} F_{\mathbf{x}_i,t}(\boldsymbol{\omega}) e^{-i\cdot\Phi(\mathbf{v})}. \tag{4}$$

We have the following decomposition:

$$\begin{aligned} F_{t+1}(\boldsymbol{\omega}) &= \sum_{\mathbf{x}_i \in \mathcal{I}} F_{\mathbf{x}_i,t+1}(\boldsymbol{\omega}) = \sum_{\mathbf{x}_i \in \check{\mathcal{B}}_t} F_{\mathbf{x}_i,t}(\boldsymbol{\omega}) e^{-i\cdot\Phi(\mathbf{v})} + \sum_{\mathbf{x}_i \in \mathcal{I} - \hat{\mathcal{B}}_{t+1}} F_{\mathbf{x}_i,t+1}(\boldsymbol{\omega}) \\ &= F_t(\boldsymbol{\omega}) e^{-i\cdot\Phi(\mathbf{v})} - \sum_{\mathbf{x}_i \in \mathcal{I} - \check{\mathcal{B}}_t} F_{\mathbf{x}_i,t}(\boldsymbol{\omega}) e^{-i\cdot\Phi(\mathbf{v})} + \sum_{\mathbf{x}_i \in \mathcal{I} - \hat{\mathcal{B}}_{t+1}} F_{\mathbf{x}_i,t+1}(\boldsymbol{\omega}). \end{aligned}$$

Although it seems impossible to calculate $\Phi(\mathbf{v})$ without the foreground/background partition, in the next section we show that good approximations of phase discrepancy is achievable in some cases.

2.2 Approximating the Phase Discrepancy

Since it is impossible to quantify the appearance and location of the pixels in $\mathcal{I} - \check{\mathcal{B}}_t$, we assume $F_{\mathbf{x}_i,t}(\boldsymbol{\omega})$ follows an independent normal distribution in the complex domain, that is:

$$\text{Real}\{F_{\mathbf{x}_i,t}(\boldsymbol{\omega})\} \sim N(0,1); \qquad \text{Imag}\{F_{\mathbf{x}_i,t}(\boldsymbol{\omega})\} \sim N(0,1). \tag{5}$$

For a simpler notation, we define a complex variable $z_i = F_{\mathbf{x}_i,t}(\boldsymbol{\omega})$. Let $Z_n = \sum_{i=1}^{n} z_i$ be the sum of this sequence. We have the following:

$$\begin{aligned} \text{Real}\{Z_n\} &\sim N(0,n) \\ \text{Imag}\{Z_n\} &\sim N(0,n) \end{aligned}$$

Because $|Z_n| = \sqrt{\text{Real}\{z_i\}^2 + \text{Imag}\{z_i\}^2}$, it follows a χ distribution with 2 degrees of freedom:

$$p(|Z_n| = x) = \sqrt{n}\sigma x e^{-x^2/2}. \tag{6}$$

Thus, the expectation of the spectral amplitude is determined by the number of pixels in the summation. More specifically:

$$\frac{E(|F_t(\boldsymbol{\omega})|)}{E(|\sum_{\mathbf{x}_i \in \check{\mathcal{B}}_t} F_{\mathbf{x}_i,t}(\boldsymbol{\omega})|)} = \frac{\sqrt{\#(\mathcal{I})}}{\sqrt{\#(\check{\mathcal{B}}_t)}}. \tag{7}$$

The number of pixels in the foreground and background are estimated from our hand labeled database (see Section 3). On average, our bounding box of the

foreground (an over-estimation of the actual foreground) occupies 5% pixels of the frame [2]. Thus we approximate the phase discrepancy in Eq.5 by:

$$\tilde{\Phi}(\mathbf{v}) = \angle F_{t+1}(\boldsymbol{\omega}) - \angle F_t(\boldsymbol{\omega}). \tag{8}$$

The estimation error comes from the pixels of the foreground and occluded parts of the background. The cumulative effect of these pixels at frequency $\boldsymbol{\omega}$ can be considered as added noise to variable η to the original variable $F_t(\boldsymbol{\omega})e^{-i\cdot\Phi(\mathbf{v})}$ in Eq.5, where:

$$\eta = -\sum_{\mathbf{x}_i \in \mathcal{I}-\check{\mathcal{B}}_t} F_{\mathbf{x}_i,t}(\boldsymbol{\omega})e^{-i\cdot\Phi(\mathbf{v})} + \sum_{\mathbf{x}_i \in \mathcal{I}-\hat{\mathcal{B}}_{t+1}} F_{\mathbf{x}_i,t+1}(\boldsymbol{\omega}).$$

From Eq.7, we set $F_t(\boldsymbol{\omega})e^{-i\cdot\Phi(\mathbf{v})}$ to 1 to determine the distribution of η:

$$E(|\eta|) = \frac{\sqrt{2\#(\mathcal{I}-\check{\mathcal{B}}_t)}}{\sqrt{\#(\check{\mathcal{B}}_t)}} \approx \sqrt{0.1}; \qquad \angle\eta \sim U(0, 2\pi). \tag{9}$$

The upper bound of error in $\tilde{\Phi}(\mathbf{v})$ is therefore:

$$\max\left[\Phi(\mathbf{v}) - \tilde{\Phi}(\mathbf{v})\right] = \max\left\{\tan^{-1}\left[E(|\eta|)\right]\right\} \approx 0.31. \tag{10}$$

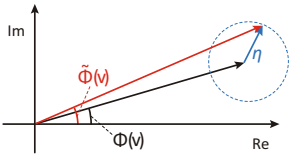


Fig. 2. A diagram of the angular error calculation. Given $E(|\eta|) = \sqrt{0.1}$, the upper bound of the angular error is 0.31 (17.6°), the mean angular error is 0.21 (12.3°).

As long as the approximation in Eq.8 holds, we can construct the estimated spectrum $\tilde{F}_{t+1}(\boldsymbol{\omega})$ from $F_t(\boldsymbol{\omega})$ and $\tilde{\Phi}$:

$$\begin{aligned}\tilde{F}_{t+1}(\boldsymbol{\omega}) &= F_t(\boldsymbol{\omega})e^{-i\cdot\tilde{\Phi}(\mathbf{v})} = |F_t(\boldsymbol{\omega})| \cdot e^{-i[\angle F_t(\boldsymbol{\omega}) + \tilde{\Phi}(\mathbf{v})]} \\ &= |F_t(\boldsymbol{\omega})| \cdot e^{-i[\angle F_{t+1}(\boldsymbol{\omega})]}\end{aligned}$$

Finally, the saliency map has the simple form:

$$\begin{aligned}s(\mathbf{x}, t) &= \left\{\mathcal{F}^{-1}\left[F_{t+1}(\boldsymbol{\omega})\right] - \mathcal{F}^{-1}\left[\tilde{F}_{t+1}(\boldsymbol{\omega})\right]\right\}^2 \\ &= \left\{\mathcal{F}^{-1}\left[\left(|F_{t+1}(\boldsymbol{\omega})| - |F_t(\boldsymbol{\omega})|\right) \cdot e^{-i\angle F_{t+1}(\boldsymbol{\omega})}\right]\right\}^2\end{aligned} \tag{11}$$

[2] In other databases such as [13] and [14], objects are in a similar size.

2.3 Eliminating Boundary Effects

The 2-D Discrete Fourier Transform implicitly implies periodicity of the signal. This property invalidates Eq.1 since pixels around the edge of the frame do not have their correspondences in the next frame. As a result, these frame-edges often have very large reconstruction errors and mislead the saliency maps (see Fig.3.C).

Assume we have two adjacent image frames. We use $\mathcal{C}_1$ and $\mathcal{C}_2$ to denote the pixels that lead to boundary effects. That is:

$$\mathcal{C}_1 = \{\mathbf{x}_i \mid \mathbf{x}_i \in \mathcal{B}_1, \mathbf{x}_i + \mathbf{v} \notin \mathcal{I}\}; \qquad \mathcal{C}_2 = \{\mathbf{x}_i \mid \mathbf{x}_i \in \mathcal{B}_2, \mathbf{x}_i - \mathbf{v} \notin \mathcal{I}\} \tag{12}$$

If we predict frame 2 based on frame 1 (as Eq.11 states), we will have a large error at $\mathcal{C}_1$. However, using Eq.11 we have no problem in recovering pixels in $\mathcal{C}_2$. Reciprocally, if we reverse the temporal order – reconstructing frame 1 from frame 2, only $\mathcal{C}_2$ has boundary effect.

In a more rigid format, we denote the temporally ordered saliency map that compares the predicted frame 2 with observed frame 2 as $\overrightarrow{s}(\mathbf{x}, t)$, and the saliency map using reversed sequence (predicting frame 1 from frame 2) as $\overleftarrow{s}(\mathbf{x}, t+1)$. We have:

$$\begin{aligned} &\overrightarrow{s}(\mathbf{x}_i, t) > \varepsilon, \quad \text{where } \mathbf{x}_i \in \mathcal{C}_1; \qquad &&\overleftarrow{s}(\mathbf{x}_i, t+1) \le \varepsilon, \quad \text{where } \mathbf{x}_i \in \mathcal{C}_1 \\ &\overrightarrow{s}(\mathbf{x}_i, t) \le \varepsilon, \quad \text{where } \mathbf{x}_i \in \mathcal{C}_2; \qquad &&\overleftarrow{s}(\mathbf{x}_i, t+1) > \varepsilon, \quad \text{where } \mathbf{x}_i \in \mathcal{C}_2, \end{aligned}$$

where ε is bounded by Eq.10.

In an elegant form, we finally eliminate the boundary effect by combining the two maps:

$$s(\mathbf{x}, t) = \sqrt{\overrightarrow{s}(\mathbf{x}, t) \cdot \overleftarrow{s}(\mathbf{x}, t+1)} \tag{13}$$

For $\forall \mathbf{x}_i \in \mathcal{C}_1 \bigcup \mathcal{C}_2$, it is easy to see that $s(\mathbf{x}_i, t) \to 0$ as either $\overrightarrow{s}(\mathbf{x}_i, t) \to 0$, or $\overleftarrow{s}(\mathbf{x}, t+1) \to 0$. The saliency map generated by Eq.13 is shown in Fig.3-D.

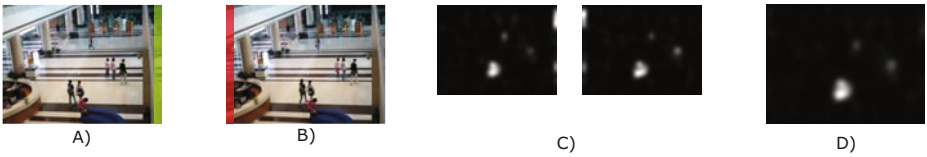


Fig. 3. An illustration of the boundary effect. **A**) & **B**): Two adjacent frames. Green and red shadows in each frame indicates $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively. **C**): The saliency map based on single sided temporal order. Note that the border effect is as strong as the moving pedestrian in the center. **D**): The final saliency map.

3 Experiments

3.1 Implementing the Phase Discrepancy Algorithm

In MATLAB, the phase discrepancy algorithm is:

```
FFT1=fft2(Frame1);
FFT2=fft2(Frame2);
Amp1=abs(FFT1);
Amp2=abs(FFT2);
Phase1=angle(FFT1);
Phase2=angle(FFT2);
mMap1=abs(ifft2((Amp2-Amp1).*exp(i*Phase1)));
mMap2=abs(ifft2((Amp2-Amp1).*exp(i*Phase2)));
mMap=mat2gray(mMap1.*mMap2);
```

`Frame1` and `Frame2` are consecutive frames. In our experiment, the size of image is gray-scaled and shrank to 120×160. On a 2.2GHz Core 2 Duo personal computer, this code performs at refresh rates as high as 75 frames per second.

One natural way to extend this algorithm to color images is to process each color channel separately, and combine saliency maps for each channel linearly. However, by tripling computational cost, the foreground pixels of color images does not seem to violate the intensity constancy assumption three times stronger than the gray-scale image. Indeed, our observation is corroborated by experiments. A comparison experiment of color image detection is in Section 3.3. Since our algorithm emphasizes processing speed, we use gray scale images in most of our experiments.

We also notice that in real world scenes, the intensity constancy assumption is subject to noises, such as background perturbation (moving leaves of a tree), sampling alias, or CCD noise. One way to reduce such noise is to combine the results from adjacent frames. However, we can only do so when the sampling rate is high enough such that the object motion in the saliency map is tolerable. In our experiments, we produce a reliable saliency map from 5 consecutive frames. At 20Hz, 5 frames takes about 0.25 second, this approach reduces the noise effectively without causing a drift in the salient region (see Fig. 4).

Fig. 4. A comparison of combining the saliency maps of different frames. **A**):One frame of one video clip. **B**): The saliency map computed by 2 frames. **C**): The saliency map by combining 5 frames (0.25 second). **D**): The saliency map by combining 20 frames (1 second).

3.2 A New Database for Moving Object Detection

There are several public databases for evaluating motion detectors and trackers, such as PETS [13] and CAVIAR [14]. However, very few of them considered camera motion. In this section we introduce a new database to evaluate the performance of an moving object detection algorithm.

Fig. 5. Sample frames of clips in the database of object motion detection. Both scenes and moving objects vary from clips to clips.

Our database consists of indoor/outdoor scenes (see Fig.5). All clips were collected by a moving video camera under 20 FPS sampling rate. Different categories of objects are included in the video clip, such as walking pedestrians, cars and bicycles, and sports players. Given the high refresh rate, motion in adjacent frames are very similar. Therefore it is unnecessary to label every frame. The original 20 FPS videos are given to our subjects for motion detection. For labeling, we asked each subject to draw bounding boxes on a small number of key frames by sub-sampling the sequence on a 0.5-second interval. Eleven naïve subjects labeled all moving objects in the video. Some numbers from this database are in Table.1.

Table 1. A summary of our database

Items	Clips	Frames	Labelers	Key frames	Bounding boxes
Number	20	2557	11	297	4785

The evaluation metric of the database is the same as PETS [15]. Although we have data from multiple subjects, the output of an algorithm is compared to one individual at a time. Let R_{GT} denotes the ground truth from the subject. The result generated by the algorithm is denoted as R_D. A detection is considered a true positive if:

$$\frac{Area(R_{GT} \cap R_D)}{Area(R_{GT} \cup R_D)} \geq Th, \tag{14}$$

The threshold Th defines the tolerance of a post-system that is connected to an object detector. If we use a loose criterion (Th is small) even a minimal overlap between the generated bounding box and ground truth is considered a success. However, for many applications, a much higher overlap, equivalent to a much tighter criterion and a larger value of Th, is needed. In our experiments, we use $Th = 0.5$.

For the n^{th} clip, using the i^{th} subject as the ground truth, we use GT_n^i, TP_n^i, FP_n^i to denote the number of ground truth, true positive, and false positive bounding boxes, respectively. The Detection Rate(DR) and False Alarm Rate (FAR) is determined by:

$$DR_n = \frac{\sum_i TP_n^i}{\sum_i GT_n^i} \qquad FAR_n = \frac{\sum_i FP_n^i}{\sum_i TP_n^i + FP_n^i}. \tag{15}$$

In a frame where multiple bounding boxes are presented, finding the correct correspondence for Eq.14 can be very hard. Given a test bounding box, we simply compare it against every ground truth bounding box, and pick the best match. Although this scheme does not guarantee that one ground truth bounding box is used only once, in practice, confusions are rare.

3.3 Performance Evaluation

To determine bounding boxes from the saliency map, an algorithm needs to know certain parameters such as spatial scale and sensitivity. To achieve a good performance without being trapped by parameter tuning, we use Non-Maximal Suppression [16] to localize the bounding boxes from the saliency map. This algorithm has three parameters $[\theta_1, \theta_2, \theta_3]$.

First, the algorithm finds all local maxima within a radius θ_1. Every local maximum greater than θ_2 is selected as the seed of a bounding box. Then, the saliency map is binarized by threshold θ_3. Finally, the rectangular contour that encompasses the white region surrounding every seed is considered as a bounding box.

It is straightforward to assume that the parametrization is consistent over different clips in our database, and the locations of objects are independent among different clips. Therefore, we use cross-validation to avoid over-fitting the model. In each iteration, we take 19 clips as the training set to find the parameters that maximizes:

$$\sum_{m \in \{training\}} DR_m(1 - FAR_m),$$

And use the remaining clip to test the performance. The final results of DR and FAR are the average among different clips. Samples of detected objects are shown in Fig.6. The quantitative result of our model is listed in Table 2.

3.4 Comparison to Previous Methods

To evaluate the performance of our algorithm, four representative algorithms are introduced to give comparative results on our database: the Mixture of Gaussian model [1], the Dynamic Visual Attention model [17], the Bayesian Surprise model [18], and the Saliency model [11]. MATLAB/C++ implementation of all these algorithms are available on authors' websites. Examples of the generated saliency maps are shown in Fig.7. As for the quantitative experimental part, the

parameters of Non-Maximal Suppression is trained in the same way as we described in Section 3.3 to generate bounding boxes from the saliency maps. The quantitative results are shown in Table 2. Our phase discrepancy model is the best in detecting moving objects.

It is worth noting that not all of these algorithms are designed to detect moving objects in a dynamic scene. In fact, the performance of an algorithm is determined by how well its underlying hypothesis is consistent with the data. In our database, an "object" is defined by its motion in contrast to the background. There is no assumption such as objects possessing unique feature, or background being monotonous. Therefore, it is not surprising that some algorithms did not perform very well in this experiment.

3.5 Database Consistency

The motivation behind the analysis of database consistency comes from the fact there is no objective "ground truth" for moving object detection. Although ground truth consistency issue is not widely concerned in the object detection and tracking databases, List *et.al.* [19] analyzed the statistical variation in the hand label data of CAVIAR [14], and showed that inter-subject variability can compromise benchmark results. In our database, we also observed that the same video clip can be interpreted in different ways. For instance, in Fig.8.A, some subjects label multiple players as one group, yet other subjects label every individual as one object.

A good benchmark should have consistent labels across subjects. To evaluate the consistency of our database, we assess the performance of the i^{th} subject

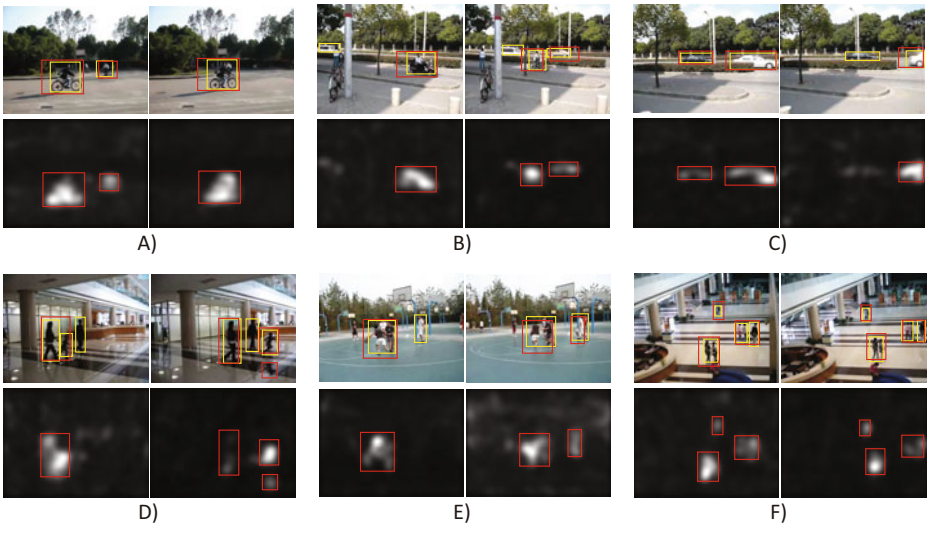


Fig. 6. Result saliency maps and the bounding boxes. In each image/saliency map pair, red bounding boxes are generated by our algorithm. Yellow bounding boxes are the ground truth drawn by a human subject.

Fig. 7. Saliency maps generated by different algorithms. **A**): Original image. **B**): Our model. **C**): Dynamic Visual Attention [17]. **D**): Bayesian Surprise [18]. **E**): Saliency [11]. **F**): Mixture of Gaussian [1].

Table 2.

	Detection Rate	False Alarm Rate
Human average	0.84 ± 0.08	0.15 ± 0.08
Our model	0.46 ± 0.14	0.58 ± 0.24
Our model (color)	0.48 ± 0.18	0.57 ± 0.24
Dynamic Visual Attention [17]	0.32±0.22	0.86±0.10
Bayesian Surprise [18]	0.12±0.09	0.92±0.04
Saliency [11]	0.09±0.08	0.98±0.01
Mixture of Gaussian [1]	0.00±0.00	1.00±0.00

based on the j^{th} subject's ground truth. Therefore, for each individual we have 10 points on the FAR-DR plot. As a comparison, the performance of our algorithm is also provided. Each data point is generated by selecting one individual as the ground truth and perform cross-validation over 20 trials. The result is shown in Fig.8.

From these results we see that even a human subject cannot achieve perfect detection. In other words, a computer algorithm is "good enough" if its performance has the same distribution as humans' on the FAR-DR plot.

Threshold and Accuracy Tolerance. Note in Eq.14, the choice of $Th = 0.5$ is arbitrary. This parameter determines the detection tolerance. To evaluate Th's influence, FAR and DR are computed as functions of Th (see Table 3).

The Influence of Object Sizes. As we have shown in Eq.10, the upper bound of error is a function of object size. To provide a empirical validation of our algorithm performance on large objects, we selected 2 clips in our database that contains the biggest objects, and tested our algorithm. The average area of the foreground objects is 10% of the image size (comparing to 5% of the original experiment). The new performance is shown in Table 4.

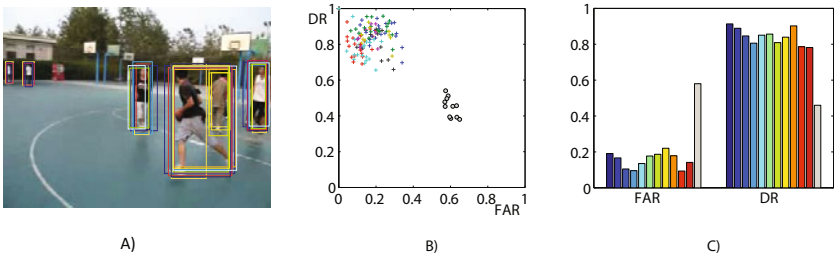


Fig. 8. A): Different interpretations of moving objects by different subjects. This image overlays the bounding boxes of 11 subjects. Boxes in the same color are drawn by the same person. We see that the incongruence among different subject is not negligible. **B**): The FAR-DR plot of all subjects and our algorithm. Each + in the same color represents the assessment of the same subject. Each ∘ indicates the performance of our algorithm. Among different subjects the DR fluctuates from 0.65 to 1, whereas the FAR fluctuates from 0 to 0.4. The average human performance is $FAR = 0.15 \pm 0.08$, $DR = 0.84 \pm 0.08$.**C**): Color bars indicate the FAR and DR for the subjects. The gray bars is the performance of our algorithm.

Table 3. Human average (DR,FAR) and model average (DR,FAR) with respect to threshold

Th	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Human Detection Rate	0.92	0.91	0.91	0.90	0.88	0.84	0.77	0.62	0.37	0.15	0.00
Human False Alarm Rate	0.07	0.07	0.07	0.08	0.11	0.15	0.22	0.37	0.62	0.85	1.00
Model Detection Rate	0.83	0.82	0.80	0.75	0.63	0.46	0.20	0.07	0.02	0.00	0.00
Model False Alarm Rate	0.18	0.20	0.24	0.31	0.43	0.58	0.80	0.93	0.98	1.00	1.00

4 Discussion and Future Work

4.1 Sources of Errors

One of the challenges is to estimate the bounding boxes for adjacent, sometimes occluded objects that move in the same direction (such as in Fig.6F). To unravel the complexity of multiple moving objects, either long term tracking, or a more powerful segmentation from saliency map to bounding boxes is required.

In some cases, we also need to incorporate top-down modulations from a level of object recognition. Since the saliency map is a pixel based representation, it favors moving parts of an object (such as a waving hand) over the entire object. A canonical interesting example is in Fig.6D: our algorithm identifies the reflection on the floor as an object. Yet none of our subjects labeled the reflection as an object.

4.2 Connections to Spectral Residual

In 2007, Hou *et.al.* proposed an interesting theory called the Spectral Residual [20]. This algorithm uses the Fourier transform of a single image to generate

Table 4. The algorithm performance over large object database. The performance drop is small.

	Original experiment	Clips with large objects
Detection Rate	0.46 ± 0.14	0.41 ± 0.14
False Alarm Rate	0.58 ± 0.24	0.65 ± 0.08

the saliency map of the static scene. As a follow-up paper suggests [21], the actual formulation of the Spectral Residual algorithm is to take the phase part of the spectrum of an image, and do the inverse transform. In other words, the saliency map generated by the Spectral Residual is the asymptotic limit of Phase Discrepancy when the second frame has $\mathbf{v} \to 0^+$. However, $\mathbf{v} \to 0^+$ is ill-defined in our problem, as the displacement approaches infinitesimal, no motion information will be available. To fully unveil the connections between these two algorithms, further research on the statistical properties of natural images is necessary.

4.3 Concluding Remarks

In this paper, we propose a new algorithm for motion detection with a moving camera in the Fourier domain. We define a new concept named Phase Discrepancy to explore camera motions. The spectrum energy of an image is generally dominated by its background. Using this, we derive an approximation to the phase discrepancy. A simple motion saliency map generation algorithm is introduced to detect moving foreground regions. The saliency map is constructed by the Inverse Fourier Transform of the difference of two successive frames spectrum energies, keeping the phase of two images invariant. The proposed algorithm does not rely on prior training on a particular feature or categories of an image. A large number of computer simulations are performed to show the strong performance of the proposed method for motion detection.

Acknowledgement

The work was supported by the National Natural Science Foundation of China (Grant No. 60775007, 90920014).

References

1. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, vol. 2, pp. 246–252 (1999)
2. Mittal, A., Paragios, N.: Motion-based background subtraction using adaptive kernel density estimation. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, vol. 2 (2004)

3. Cheung, S., Kamath, C.: Robust techniques for background subtraction in urban traffic video. In: Video Communications and Image Processing, SPIE Electronic Imaging, vol. 5308, pp. 881–892 (2004)
4. Viola, P., Jones, M.: Robust real-time face detection. International Journal of Computer Vision 57, 137–154 (2004)
5. Dollár, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: A benchmark. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 304–311 (2009)
6. Tian, T., Tomasi, C., Heeger, D.: Comparison of approaches to egomotion computation. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 315–320 (1996)
7. Han, M., Kanade, T.: Reconstruction of a scene with multiple linearly moving objects. International Journal of Computer Vision 59, 285–300 (2004)
8. Irani, M., Anandan, P.: A unified approach to moving object detection in 2D and 3D scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence 20, 577–589 (1998)
9. Vernon, D.: Fourier vision: segmentation and velocity measurement using the Fourier transform. Kluwer Academic Publishers, Dordrecht (2001)
10. Black, M., Anandan, P.: A framework for the robust estimation of optical flow. In: Proc. IEEE Conf. on International Conference of Computer Vision, pp. 231–236 (1993)
11. Itti, L., Koch, C., Niebur, E., et al.: A model of saliency-based visual attention for rapid scene analysis. IEEE Trans. on Pattern Analysis and Machine Intelligence 20, 1254–1259 (1998)
12. Mallat, S.: A wavelet tour of signal processing. Academic Press, London (1999)
13. http://ftp.pets.rdg.ac.uk
14. http://homepages.inf.ed.ac.uk/rbf/caviar/
15. Bashir, F., Porikli, F.: Performance evaluation of object detection and tracking systems. In: IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2006) (2006)
16. Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis, and Machine Vision. Cengage-Engineering (2007)
17. Hou, X., Zhang, L.: Dynamic Visual Attention: Searching for coding length increments. In: Advances in Neural Information Processing Systems, vol. 21, pp. 681–688 (2008)
18. Itti, L., Baldi, P.: Bayesian surprise attracts human attention, pp. 547–554 (2006)
19. List, T., Bins, J., Vazquez, J., Fisher, R.: Performance evaluating the evaluator. In: Proc. IEEE Joint Workshop on Visual Surveillance and Performance Analysis of Video Surveillance and Tracking (2005)
20. Hou, X., Zhang, L.: Saliency detection: A spectral residual approach. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2007), pp. 1–8. IEEE Computer Society, Citeseer (2007)
21. Guo, C., Ma, Q., Zhang, L.: Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2008)

Image Classification Using Spatial Pyramid Coding and Visual Word Reweighting

Chunjie Zhang[1], Jing Liu[1], Jinqiao Wang[1], Qi Tian[2], Changsheng Xu[1], Hanqing Lu[1], and Songde Ma[1]

[1] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, P.O. Box 2728, Beijing, China
{cjzhang,jliu,jqwang,csxu,luhq}@nlpr.ia.ac.cn, masd@most.cn
[2] University of Texas at San Antonio, One UTSA Circle San Antonio Texas, 78249-USA
qitian@cs.utsa.edu

Abstract. The ignorance on spatial information and semantics of visual words becomes main obstacles in the bag-of-visual-words (BoW) method for image classification. To address the obstacles, we present an improved BoW representation using spatial pyramid coding (SPC) and visual word reweighting. In SPC procedure, we adopt the sparse coding technique to encode visual features with the spatial constraint. Visual features from the same spatial sub-region of images are collected to generate the visual vocabulary. Additionally, a relaxed but simple solution for semantic embedding into visual words is proposed. We relax the semantic embedding from ideal semantic correspondence to naive semantic purity of visual words, and reweight each visual word according to its semantic purity. Higher weights are given to semantically distinctive visual words, and lower weights to semantically general ones. Experiments on a public dataset demonstrate the effectiveness of the proposed method.

Keywords: spatial pyramid coding, bag-of-visual-words (BoW), reweighting, image classification.

1 Introduction

In recent years, the bag-of-visual-words (BoW) model becomes popular in image classification. This model extracts appearance descriptors from local patches and quantizes them into discrete "visual words", and then a compact histogram representation is used to represent images. The descriptive power of the BoW model is severely limited because it discards the spatial information of local descriptors. To overcome this problem, one popular extension method, called the *spatial pyramid matching* (SPM) by Lazebnik *et al* [1], has been shown to be effective for image classification. The SPM partitions an image into several segments in different scales, then computes the BoW histogram within each segment and concatenates all the histograms to form a high dimension vector representation of the image.

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 239–249, 2011.

To obtain good performances, researchers have empirically found that the SPM should be used together with SVM classifier using nonlinear Mercer kernels, e.g. *Chi-square kernel* or *intersection kernel*. However, the computational complexity is $O(n^3)$ and the memory complexity is $O(n^2)$ in the training phase, where n is the size of training dataset. This constrains the scalability of the SPM-based nonlinear SVM method. To reduce the training complexity, a linear spatial pyramid matching method using sparse coding (ScSPM) is proposed by Yang *et al* [2]. This method is more robust to local spatial translations and is biological plausible [3]. Inspired by this, Wang et al [4] used locality in feature space to constrain the linear sparse coding phase (LLC) of ScSPM which further reduced the computation time. However, the performance improvement of LLC over ScSPM on real world images is not obvious. In fact, there is another constraint which was neglected in [4], i.e., the spatial locality constraint. For example, 'sky' often lies on the upper side of images, while 'beach' often lies on the lower side of images. When we try to encode an image region about the upper 'sky', it is more meaningful to use the bases which are generated by the local features on the upper side of images. Similarly, it is more meaningful to encode the lower 'beach' with the bases generated from the local features on the lower side of images.

Besides, the semantic meaning of visual word has not been considered too much in literature, which has become another obstacle to affect the performance of the BoW model. Ideally, the correspondence between visual words and semantics, namely the semantic embedding into the BoW representation, will bring the more representative and discriminative description for image classification than solely on visual features. However, the well-known semantic gap becomes a natural barrier to achieve such correspondence. Some recent work appeal to various supervised learning approaches [5, 6] to learn discriminative visual vocabulary. In fact, such supervised refinement emphasizes on the discriminative abilities of visual words rather than truly embedding semantics into image representation. We believe that the semantic embedding can further enhance the discriminative ability of visual words in image classification, but not vice versa. Consequently, it is necessary to find a suitable way to obtain such a semantic embedded BoW presentation for image classification.

In this paper, we present a novel image classification method by using spatial pyramid coding (SPC) along with visual word reweighting, as shown in Figure 1. We first partition images into sub-regions on multiple scales, and adopt the sparse coding approach to encode visual features of images with the spatial constraint. Different from SPM [1], the SPC-based visual vocabulary is concatenated with each encoding results from the sub-regions which have the same spatial locality and segmentation scale. For the semantic embedding, we adopt a relaxed but simple solution to reweight the SPC-based BoW representation according to the semantic purity of each visual word, instead of the obtainment of the semantic correspondence. Specifically, we give higher weights to semantically distinctive visual words, and lower weights to semantically general visual words.

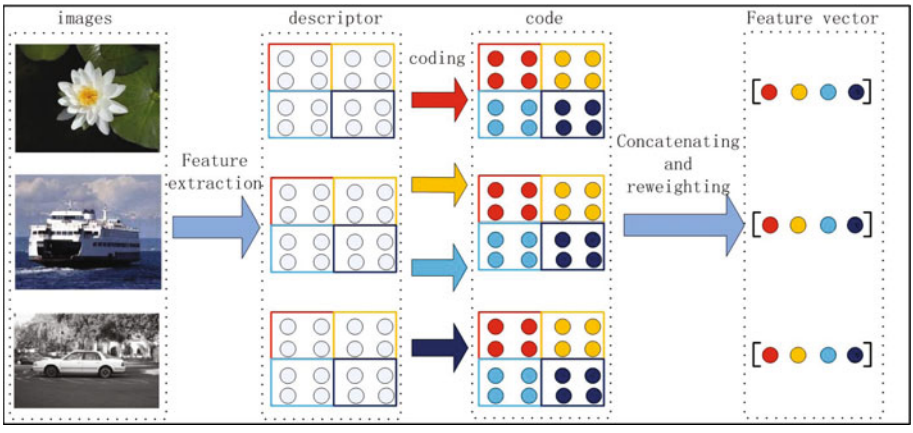


Fig. 1. Flowchart of the proposed spatial pyramid codebook (with two scales) and visual word reweighting methods. It is best viewed in color

Comprehensive experimental evaluations on the Scene-15 dataset demonstrate the effectiveness of the proposed method.

The rest of the paper is organized as follows. Section 2 gives an overview of some related work. In Section 3, we present the details of the proposed spatial pyramid coding and visual word reweighting method. Experimental results and comprehensive analysis are given in Section 4. Finally, the conclusions and future research issues are discussed in Section 5.

2 Related Work

The bag-of-visual-words model (BoW) has been widely used due to its simplicity and good performance. Many works has been done to improve the performance of the traditional bag-of-visual-words model over the past few years. Some literatures devoted to learn discriminative visual vocabulary for object recognition [7-9]. Perronnin *et al* [7] used the Gaussian Mixture Model (GMM) to perform clustering. To alleviate the drawback of k-means clustering, Jurie and Triggs [8] tried to use a scalable acceptance-radius based clustering method instead. Moosmann *et al* [9] used random forests to construct codebook which helps to improve the classification performance. Others tried to model the co-occurrence of visual words in a generative framework [10-13]. Boiman *et al* [10] tried to classify images by nearest-neighbor classification. Bosch *et al* [11] tried to classify scene images using a hybrid generative/discriminative approach. Besides, many researchers also [1, 14-19] tried to learn more discriminative classifiers by combining the spatial and contextual information of visual words. Oliva and Torralba [15] modeled the shape of the scene by using a holistic representation. Gemert *et al* [16] proposed to learn visual word ambiguity through soft assignment. Zhang *et al* [17] utilized nearest neighbor classification for visual category recognition. Motivated by Grauman and Darrell's [19] pyramid matching in feature space,

Lazebnik *et al* [1] proposed the spatial pyramid matching (SPM) which has been proven efficient for image classification.

Although the SPM method works well for image classification, it has to be used along with nonlinear Mercer kernels for good performance. However, the computational cost is $O(n^3)$ in training phase. To improve the scalability, Yang *et al* [2] proposed a linear spatial pyramid matching method using sparse coding along with max pooling to classify images, which has been shown very effective and efficient. The approach relaxes the restrictive cardinality constraint of vector quantization in traditional BoW model and uses max spatial pooling to compute histogram which reduces the training complexity to $O(n)$. Motivated by this, many researchers [4, 20-21] proposed novel methods to further improve the performance. Wang *et al* [4] proposed to use locality constraints in feature space during the sparse coding phase of [2] and the theoretical justifications are given by Yu *et al* [20]. Boureau *et al* [21] also proposed a novel method to learn a supervised discriminative dictionary for sparse coding.

Obviously, not all of the visual words are equally useful for image classification. [22-23] showed that the human visual system employs an effective attention mechanism and can recognize different object categories robustly by focusing on the interesting parts in an image. To choose the most discriminative visual features, Liu *et al* [24] tried to select the most discriminative visual word combination with Adaboost while Mutch and Lowe [25] used sparse, localized features for multiclass object recognition. Cai *et al* [26] also tried to learn weights for each visual word by solving a quadratic programming problem.

3 Spatial Pyramid Coding and Visual Word Reweighting

This section gives the details of the proposed spatial pyramid coding (SPC) and visual word reweighting method. For each image, we first densely extract local image features and then utilize the spatial pyramid principle to encode local features. Then we concatenate the BoW representation of different segments and reweight each visual word based on its semantic purity. Figure 1 shows the flowchart of the proposed spatial pyramid coding and visual word reweighting method.

3.1 Spatial Pyramid Coding

The idea of using spatial pyramid along with the BoW representation of images has been proven very effective for image classification by many researchers. This method partitions an image into increasingly finer spatial sub-regions and computes the histogram of local features from every sub-region [1]. Usually, $2^l \times 2^l$ subregions, with $l = 0, 1, 2$ are used. Other partition method such as 3×1 is also used to incorporate top and bottom relationships, which has been proven very useful on the PASCAL VOC Challenge. Take the $2^l \times 2^l$ for example, for L levels and M channels, the resulting concatenated vector for each image has a dimensionality of $M \sum_{l=0}^{L} = M\frac{1}{3}(4^{L+1} - 1)$.

To preserve the discriminative power of local image features as much as possible, researchers have tried many coding methods, among which the most popular is the k-means model. Formally, let X be a set of D-dimensional local features. The number of local features is N, i.e. $X = [x_1, x_2, ..., x_N] \in R^{D \times N}$ where $x_i \in R^{D \times 1}$. Suppose we have a codebook B with M visual words, where $B = [b_1, b_2, ..., b_M] \in R^{D \times M}$. To convert each descriptor into a M-dimensional vector to represent images, k-means based vector quantization (VQ) method tries to solve a constrained least square fitting problem as:

$$C = argmin \sum_{i=1}^{N} = \| x_i - B \times c_i \|^2 \tag{1}$$

$$s.t. \| c_i \|_{l^0} = 1, \| c_i \|_{l^1} = 1, c_{ij} \geq 0, \forall i, j$$

where $C = [c_1, c_2, ..., c_N]$ is the codes for X and c_{ij} is the j-th element of c_i.

The constraints in the k-means model are very restrictive with only one element of c_i is set to 1. In practice, this is often achieved by nearest neighbor search. To alleviate the discriminative power loss during vector quantization, Yang *et al* [2] proposed to use sparse coding instead. They relaxed the restrictive cardinality constraint in Eq. (1) by using a sparsity regularization term instead.l^1 norm of c_i is used. Thus, Eq. (1) becomes a standard sparse coding problem [27] as:

$$C = argmin \sum_{i=1}^{N} = \| x_i - B \times c_i \|^2 + \lambda \| c_i \|_1 \tag{2}$$

where λ is the regularization parameter and $\| \cdot \|_1$ is the l^1 norm which sums the absolute value of each element. This can be solved by optimizing over each individually.

However, as introduced in [4], locality is more essential than sparsity because locality leads to sparsity but not necessary vice versa. It allows sparse reconstruction of features in the appearance space using sparsity along with locality constraints. However, this discards the spatial information in the coding phase. This paper proposes an "orthogonal" approach: we perform pyramid coding in the two-dimensional image space and use sparse coding method [1, 27] in feature space. Specifically, we first partition the image into increasingly finer spatial subregions with $2^l \times 2^l$, $l = 0, 1, 2$. For each sub-region, the sparse coding parameters and the codebook are then jointly learned using the local image features within this sub-region. This is achieved by alternatively optimizing over the codebook B and the coding parameters C while keeping the other fixed. We use the alternative optimization method as did in [1, 27] to solve this problem. In our experiments, about 45,000 SIFT descriptors extracted from random patches of each segment are used to train the codebooks. Once we have learned the codebook for each sub-region, we are able to code efficiently for each local feature using Eq. (2). Max pooling [1] is then used to generate the BoW representation for each segment which has been shown very effective when combined with sparse

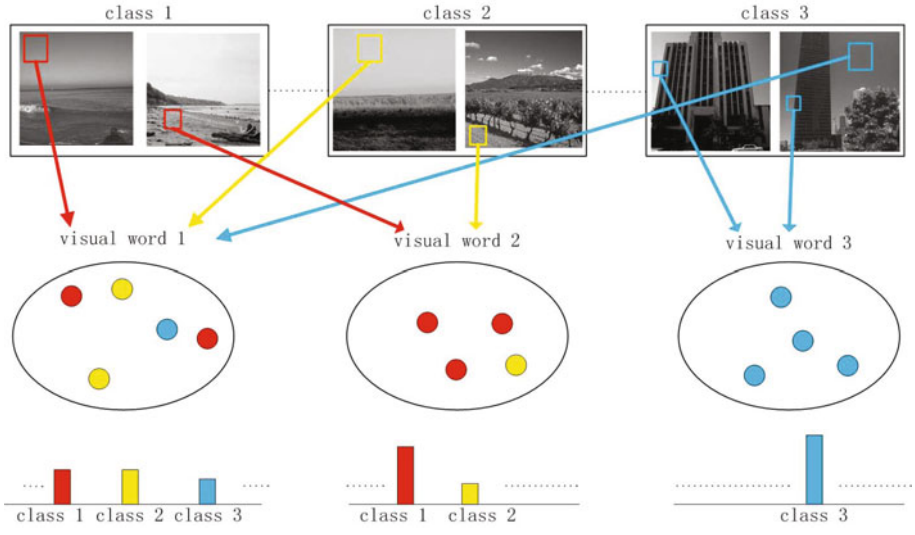


Fig. 2. Toy example showing the semantic meaning of visual words. Different colors represent local features extracted from different classes. Since visual word 3 is the most semantically distinctive, we believe the word is more discriminative than visual word 1 and 2 in a specific classification task. It is best viewed in color.

coding. Finally, the BoW representations of all segments are concatenated into a long vector to represent images.

3.2 Visual Word Reweighting

Although the bag-of-visual-words model is inspired by the bag-of-words approach to text categorization, the semantic meaning of visual word has not been considered too much in literature. We believe the semantic information of visual words can also be utilized to improve the image classification performance.

During the vector quantization of traditional BoW model or the sparse coding process, many local features are assigned to one visual word. These local features may come from different classes of images hence have different semantic meanings. Assuming each local image feature having the same semantic label as the image from which it is extracted, we can use the frequency distribution of classes of local features assigned to each visual word to represent this visual word. Formally, let $Q = [q_1, q_2, ..., q_M] \in R^{K \times M}$ is the semantic distribution of all the visual words, where $q_i \in R^{K \times 1}$ and K is the number of classes. We believe that the purity of each visual word is correlated with its discriminative power. For example, sky often exists on the outdoor scene images. While classifying outdoor images of different classes, visual words representing the upper sky are often generated by local features extracted from different classes of images. These visual words are noisy for classification and should be given lower weights. On the contrary, if one visual word is generated mainly by the local features of

the same class, the discriminative power of this visual word is much stronger than visual words which are generated by local features from diverse classes of images. Figure 2 shows a toy example reflecting showing the semantic purity of visual words.

To measure the semantic purity of each visual word quantitatively, we choose to use the entropy of each visual word's semantic distribution, because it has been proven very effective and efficient to implement. The larger the entropy, the less pure the visual word and vice versa. Formally, let e_i to represent the entropy of visual word b_i whose semantic distribution is q_i. e_i can then be calculated as:

$$e_i = -\sum_{k=1}^{K} q_{ik} ln(q_{ik}) \tag{3}$$

Let w_i to represent the weight of visual word i,$i \in 1, 2, ..., M$. The weight of each visual word can then be computed as:

$$w_i = exp(-e_i/\alpha) \tag{4}$$

where α is the scaling parameter. In our experiments, we simply set α to 1. The weight of each visual word can then be computed in an efficient way as:

$$w_i = \prod_{k=1}^{K} q_{ik}^{q_{ik}} \tag{5}$$

4 Experiments

We evaluate the proposed spatial pyramid coding and visual word reweighting method on the fifteen natural scene dataset by provided Lazebnik *et al* [1]. The fifteen scene dataset composes 4,485 images, which vary from natural scenes like forests and mountains to man-made environments like offices and kitchens. Thirteen were provided by Fei-Fei and Perona [12] (eight of these were originally provided by Oliva and Torralba [15]) and two were collected by Lazebnik *et al* [1]. We perform all processing in grayscale of images even when sometimes the color images are provided. As to the feature extraction, we follow Lazebnik *et al* [1] and densely compute SIFT descriptors on overlapping 16×16 pixels with an overlap of 8 pixels. The codebook size is set to 1,024, as Yang *et al* [2] did. Multi-class classification is done via the one-versus-all rule: a SVM classifier is learned to separate each class from the rest and a test image is assigned the label of the classifier with the highest response. The average of per-class classification rates is used to quantitatively measure the performance.

We show some example images of the Scene-15 dataset in Figure 3. The major picture sources in this dataset include the COREL collection, personal photographs and Google image search. Each category has 200 to 400 images, and the average image size is 300×250 pixels. We follow the same experiment procedure of Lazebnik *et al* [2] and randomly choose 100 images per category as

Fig. 3. Example images of the Scene-15 dataset

Table 1. Classification rate comparison on the Scene-15 dataset. Numerical values in the table stand for mean and standard derivation.

Algorithms	Classification Rate
KSPM[2]	76.73 ± 0.65
KC[16]	76.67 ± 0.39
ScSPM[2]	80.28 ± 0.93
ScSPM	78.77 ± 0.50
SPC	81.14 ± 0.46
SPC+Reweighting	$\mathbf{82.98 \pm 0.23}$

the training set and use the remaining images as the test set. This process is repeated for five times.

Table 1 gives the detailed comparison results. We compare the proposed methods with the kernel codebook proposed by Gemert *et al* [16], the ScSPM and the reimplementation of nonlinear kernel SPM by Yang *et al* [2]. Our implementation of ScSPM is not able to reproduce the results reported by Yang *et al* [2] probably due to the feature extraction process and normalization process. We can see from the results that the proposed SPC outperforms ScSPM, which shows the effectiveness of combining spatial information in the coding phase. Besides, the classification rate can be further improved by reweighting each visual word based on its semantic purity. This demonstrates the effectiveness of the proposed method.

Table 2. Classification rate per concept for the ScSPM, SPC and SPC+Reweighting

Class	ScSPM	SPC	SPC+Reweighting
Bedroom	67.24± 5.57	83.62± 1.16	**84.48± 1.28**
CALsuburb	99.29± 1.42	99.29± 0.95	99.29± 1.00
Industrial	56.40± 2.00	57.35± 2.67	**57.82± 3.22**
Kitchen	66.36± 3.44	65.45± 2.54	**69.09± 4.96**
Livingroom	62.43± 2.92	64.02± 2.55	**65.61± 3.42**
MITcoast	97.69± 1.51	96.15± 0.61	**98.08± 1.87**
MITforest	97.81± 0.91	**99.12± 1.30**	97.37± 1.00
MIThighway	86.25± 2.67	88.12± 4.34	**88.12± 3.71**
MITinsidecity	88.94± 1.16	88.94± 1.43	**89.90± 1.50**
MITmountain	84.67± 2.70	**86.50± 2.96**	85.77± 2.83
MITopencountry	74.19± 3.33	79.03± 4.55	**100± 0.00**
MITstreet	91.15± 2.29	**94.79± 3.31**	92.71± 3.01
MITtallbuilding	97.27± 0.35	98.05± 0.33	**99.22± 0.28**
PARoffice	86.96± 2.25	**87.83± 2.84**	83.48± 0.78
Store	69.77± 2.70	73.03± 3.50	**73.95± 3.59**

To analyze the detailed classification performance, we give the classification rate per concept in table 2. Generally, four conclusions can be made. First, we can have similar observation as [1] did that the indoor classes (e.g. kitchen, livingroom) are more difficult to classify than the outdoor classes (e.g. MITopencountry, MITtallbuilding). Second, the advantages of SPC over ScSPM mainly focus on indoor classes, e.g. bedroom, livingroom and store. This is because the SPC method is able to combine the spatial information into the coding process; hence helps make correct categorization of images. Third, the improvement of SPC+Reweighting over SPC mainly lies on outdoor classes, this is because images of the outdoor classes (e.g. "MITopencountry") are relative simple and with less objects compared with images of indoor classes. We believe this is the reason why the reweighting works. Finally, the proposed SPC and SPC+Reweighting methods outperform ScSPM for all the fifteen classes.

5 Conclusion

This paper proposes a novel method for image classification using spatial pyramid coding (SPC) and visual word reweighting. SPC is easy to compute and can incorporate spatial information in the coding phase which is lost in the sparse coding spatial pyramid matching (ScSPM). SPC applies spatial constraint in the coding phase for each sub-region of images; hence is more discriminative than ScSPM. Besides, we relax the semantic embedding from ideal semantic correspondence to semantic purity of visual words and reweight each visual word according to its semantic purity, giving higher weights to semantically distinctive visual words, and lower weights to semantically general ones. The experimental evaluations on the Scene-15 dataset demonstrate the effectiveness of the proposed spatial pyramid coding and visual word reweighting for image classification.

Our future work includes the following possible directions. First, More efficient coding methods, such as semi-supervised methods will be studied. Second, how to further reduce the computation cost will also be investigated. Third, how to integrate the spatial information of local features more efficiently will also be studied.

Acknowledgement. This work is supported by Major State Basic Research Development Program (2010CB327905) and the Natural Science Foundation of China (Grant No. 60835002, 60723005, 60723005).

References

1. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: Proc. CVPR (2006)
2. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: Proc. CVPR (2009)
3. Serre, T., Wolf, L., Poggio, T.: Object recognition with features inspired by visual cortex. In: Proc. CVPR (2005)
4. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong, Y.: Locality-constrained linear coding for image classification. In: Proc. CVPR (2010)
5. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Supervised Dictionary Learning. In: Proc. ECCV (2008)
6. Lazebnik, S., Raginsky, M.: Supervised learning of quantizer codebooks by information loss minimization. PAMI (2009)
7. Perronnin, F., Dance, C., Csurka, G., Bressan, M.: Adapted vocabularies for generic visual categorization. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 464–475. Springer, Heidelberg (2006)
8. Jurie, F., Triggs, B.: Creating efficient codebooks for visual recognition. In: Proc. ICCV, pp. 17–21 (2005)
9. Moosmann, F., Nowak, E., Jurie, F.: Randomized clustering forests for image classification. IEEE Trans. on Pattern Analysis and Machine Intelligence 30(9), 1632–1646 (2008)
10. Boiman, O., Shechtman, E., Irani, M.: In defense of nearest-neighbor based image classification. In: Proc. CVPR (2008)
11. Bosch, A., Zisserman, A., Munoz, X.: Scene classification using a hybrid generative/discriminative approach. IEEE Trans. on Pattern Analysis and Machine Intelligence (2008)
12. Fei-Fei, L., Perona, P.: A Bayesian hierarchical model for learning natural scene categories. In: CVPR (2005)
13. 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: WGMBV (2004)
14. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. Technical report, CalTech (2007)
15. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. IJCV 42(3) (2001)
16. Gemert, J., Veenman, C., Smeulders, A., Geusebroek, J.: Visual word ambiguity. IEEE Transactions and Pattern Analysis and Machine Intelligence

17. Zhang, H., Berg, A., Maire, M., Malik, J.: Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In: Proc. CVPR (2006)
18. Sivic, J.S., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: Proc. ICCV, vol. 2, pp. 1470–1477 (2003)
19. Grauman, K., Darrell, T.: The pyramid match kernel: discriminative classification with sets of image features. In: Proc. ICCV, pp.1458–1465 (2005)
20. Yu, K., Zhang, T., Gong, Y.: Nonlinear learning using local coordinate coding. In: Proc. NIPS (2009)
21. Boureau, Y.-L., Bach, F., LeCun, Y., Ponce, J.: Learning mid-level features for recognition. In: Proc. CVPR (2010)
22. Tsotsos, J.: Analyzing vision at the complexity level. Behav. Brain Sci. 13, 423–469 (1990)
23. Chen, X., Zelinsky, G.J.: Real-world visual search is dominated by top-down guidance. Vision Research 46, 4118–4133 (2006)
24. Liu, D., Hua, G., Viola, P., Chen, T.: Integrated feature selection and higher-order spatial feature extraction for object categorization. In: Proc. CVPR (2008)
25. Mutch, J., Lowe, D.G.: Multiclass object recognition with sparse, localized features. In: Proc. CVPR (2006)
26. Cai, H., Yan, F., Mikolajczyk, K.: Learning weights for codebook in image classification and retrieval. In: Proc. CVPR (2010)
27. Lee, H., Battle, A., Raina, R., Ng, A.: Efficient sparse coding algorithms. In: Advances in Neural Information Processing Systems, pp. 801–808. MIT Press, Cambridge (2007)
28. Zhang, C., Liu, J., Ouyang, Y., Tian, Q., Lu, H., Ma, S.: Category sensitive codebook construction for object category recognition. In: ICIP (2009)

Class-Specific Low-Dimensional Representation of Local Features for Viewpoint Invariant Object Recognition

Bisser Raytchev, Yuta Kikutsugi, Toru Tamaki, and Kazufumi Kaneda

Department of Information Engineering, Hiroshima University, Japan
{bisser,kikutsugi,tamaki,kin}@hiroshima-u.ac.jp

Abstract. In this paper we propose a new general framework to obtain more distinctive local invariant features by projecting the original feature descriptors into low–dimensional feature space, while simultaneously incorporating also class information. In the resulting feature space, the features from different objects project to separate areas, while locally the metric relations between features corresponding to the same object are preserved. The low–dimensional feature embedding is obtained by a modified version of classical Multidimensional Scaling, which we call supervised Multidimensional Scaling (sMDS). Experimental results on a database containing images of several different objects with large variation in scale, viewpoint, illumination conditions and background clutter support the view that embedding class information into the feature representation is beneficial and results in more accurate object recognition.

1 Introduction

Local invariant features like SIFT [1], SURF [2], HOG [3], etc. are becoming increasingly popular in computer vision and pattern recognition, finding numerous applications and often being the tool of choice in as diverse areas as image registration, 3D reconstruction, image retrieval, robot navigation and object recognition, to name just a few. A comprehensive survey of local invariant features is given in [4], which also provides a qualitative evaluation of their strengths and weaknesses. In the context of object recognition, especially, one inconvenience with local features is that they cannot be fed directly into a classifier, like in global appearance–based methods [5], where either the whole image, or the response of a pre–determined number of filters on the image is used as a feature. This problem stems from the fact that usually a different number of features are extracted from each image of an object, and there is no obvious way how to organize them in a vector form, suitable for input to a standard classifier. This difficulty has been surmounted to some extent in the recently popular bag–of–keypoints (BoK) framework [6], where the features extracted from all training images are clustered in feature space, and each image is represented by a histogram in which the cluster centers, also called "visual words" [7], determine the bins.

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 250–261, 2011.

However, one problem with the BoK approach is that features extracted from different objects can be clustered together, which may lead to a similar representation of different classes of objects. This makes recognition difficult and unreliable. Using a good classifier in a classification step after the representation step can partially improve the situation, but a better and more natural solution might be to take measures earlier, at the representation step, not allowing the features from different objects to be clustered together in the first place. In this way, the available class information can be used more efficiently, resulting in an easier classification problem.

Our main idea is to embed the local feature descriptors in a low–dimensional space where the features from different classes of objects are well separated from each other. Then the cluster centers for different classes will also be located widely apart from each other, and this would lead to more accurate classification in the framework of the bag–of–keypoints approach. In order to accomplish this, we propose a new low–dimensional embedding method, Supervised Multidimensional Scaling (sMDS), which is based on classical multidimensional scaling (MDS), but the low–dimensional embedding of the data is modified to reflect the available class information. Classical MDS [8], also known as Principle Coordinates Analysis, is similar to Principle Components Analysis (PCA) [9], but it operates on the distance matrix of the data, rather than on the data itself, as PCA does. In our case, we manipulate the distance matrix obtained from the data to ensure that in the low–dimensional embedding, the features corresponding to different objects are separated from each other. An additional benefit is the lower–dimensionality of the features, which can speed up significantly the clustering process in the BoK framework.

The rest of the paper is organized as follows. In section 2 we briefly review related work, concentrating more specifically on the PCA–SIFT algorithm. In section 3, the Supervised Multidimensional Scaling (sMDS) algorithm is explained, which is used to illustrate and implement the general class–specific low–dimensional feature representation framework proposed in this paper. Experimental results comparing the performance of the features obtained by using sMDS, PCA–SIFT, LDA–SIFT (an LDA version of SIFT, explained below) and SIFT in a BoK based object recognition task and a feature matching based image retrieval task are shown in section 4, and section 5 concludes the paper.

2 Related Work

Our work is most strongly related to the PCA–SIFT algorithm of Ke and Sukthankar proposed in [10]. They apply Principle Component Analysis (PCA) to the normalized gradient patches in an image, centered at locations where keypoints (interest points) have been detected by a SIFT detector. The SIFT algorithm [1], like many other algorithms for local invariant feature extraction, consists of two main parts: a feature detector and a feature descriptor. The SIFT feature detector finds interest points in an image by searching for local extrema in the scale–space pyramid built with Difference–of–Gaussian (DoG)

filters. Apart from its location, the detected keypoint is assigned also scale and dominant orientation, and these are used to build a canonical view of the local gradient of the image patch that is invariant to similarity transforms. The standard SIFT descriptor then constructs smoothed orientation histograms to represent the structure of the patch around the keypoint. In PCA–SIFT, the last step (building of the orientation histogram) is substituted by a low–dimensional projection of the normalized gradient patch using PCA. The motivation for this is to achieve a more compact (low–dimensional), faster, more accurate and theoretically simpler descriptor, avoiding the somewhat heuristic choices behind the design of the SIFT descriptor. Since the detector part of SIFT is kept unchanged, the advantages it provides in terms of good feature localization ability and repeatability are retained. In [10], the authors compare the performance of SIFT and PCA–SIFT in a number of feature matching tasks and in an image retrieval application, demonstrating that PCA–SIFT results in a more distinctive, compact and robust descriptor. The authors suggest that one of the most important reasons for the success of PCA–SIFT can be linked to the low–dimensional projection of the gradient patch, which appears to retain the identity–related variations, while discarding the distortions induced by other effects of the imaging process.

The main contribution of the present paper is to develop further the main idea of PCA–SIFT – low dimensional mapping of local invariant features – by incorporating also object class information. This would result in more distinctive features, which is highly desirable for object recognition applications. Surprisingly, in the context of local image descriptors, this direction has not attracted much attention, although quite a few works based on PCA have been reported, e.g. [11], [12]. Some previous work in the direction of trying to obtain more distinctive local features has been done in [13], where the authors suggest using Linear Discriminant Analysis (LDA) [14] to obtain low–dimensional projection of the features. However, in that work the projection is determined by only two classes – matching and non–matching pairs of features – which might be advantageous in the context of wide–baseline matching or other feature matching applications, but would be less relevant for object recognition tasks, especially in the bag–of–keypoints (BoK) framework, where many features are grouped in a single cluster, and the gain in matching precision would anyway be lost in the clustering.

In the approach proposed in the present paper, embedding of class information into the features is achieved by projecting them to a low–dimensional space, in which the features from different objects project to separate areas, while at the same time the metric relations between features corresponding to the same object are preserved. This is accomplished by using the supervised Multidimensional Scaling algorithm introduced below which directly manipulates the distance matrix of the features, thus guaranteeing (by construction) that features from different classes would map to well–separated areas in low–dimensional space. In this way, it is expected (and later experimentally verified) that the "averaging" or "blurring" effect of the feature clustering in the BoK framework can be alleviated, resulting in more precise and reliable recognition.

3 Low-Dimensional Feature Representation by Supervised Multidimensional Scaling (sMDS)

The supervised Multidimensional Scaling (sMDS) algorithm proposed in this paper is a modification of the classical (unsupervised) Multidimensional Scaling algorithm [15], [16], which is first briefly reviewed here. Multidimensional scaling is a term used to denote a group of techniques which obtain a low–dimensional representation of a set of data points by analyzing the distance matrix of the data. There are many types of MDS, but the classical MDS proceeds as follows.

From the $n \times n$ distance matrix $\boldsymbol{D} = (d_{ij})$ of the data (feature descriptors $\boldsymbol{x}_i$,$i : 1...n$, extracted from the training images in our case), the following $n \times n$ matrix is formed:

$$\boldsymbol{A} = (a_{ij}), \quad a_{ij} = -\frac{1}{2} d_{ij}^2 \tag{1}$$

Then, the "doubly centered" matrix $\boldsymbol{B}$=$\boldsymbol{HAH}$ is formed, where $\boldsymbol{H}$ is the centering matrix

$$\boldsymbol{H} = \boldsymbol{I} - n^{-1}\boldsymbol{J}_n, \boldsymbol{J}_n = \boldsymbol{1}_n \boldsymbol{1}_n^T \tag{2}$$

and $\boldsymbol{J}_n$ is an $(n \times n)$ matrix of ones. Next, the eigenvectors $\boldsymbol{v}_i$ corresponding to the t largest positive eigenvalues λ_i of $\boldsymbol{B}$ are found, and the required t-dimensional embedding of the data is given by

$$\boldsymbol{Y} = \boldsymbol{V}\boldsymbol{\Lambda}^{\frac{1}{2}} = (\sqrt{\lambda_1}\boldsymbol{v}_1, ..., \sqrt{\lambda_t}\boldsymbol{v}_t) = (\boldsymbol{y}_1, ..., \boldsymbol{y}_n)^T \tag{3}$$

$\boldsymbol{y}_i$ are called the *principle coordinates* of the original high-dimensional data $\boldsymbol{x}_i$, and their inter–point distances are equal to the corresponding distances in the original distance matrix $\boldsymbol{D}$.

In order to ensure that features coming from different objects are mapped to well–separated locations in low–dimensional space, in sMDS, the supervised version of MDS, we propose to manipulate the distance matrix $\boldsymbol{D}$ in the following manner:

$$\boldsymbol{D} := \boldsymbol{D} + \max(\boldsymbol{D})\boldsymbol{\Omega} \tag{4}$$

where $\max(\boldsymbol{D})$ is the maximum entry in $\boldsymbol{D}$, and $\boldsymbol{\Omega}$ is a matrix of the same size as $\boldsymbol{D}$, defined as $\boldsymbol{\Omega}_{ij} = |c(\boldsymbol{x}_i) - c(\boldsymbol{x}_j)|$. Here, $c(\boldsymbol{x}_i)$ returns the class number of the i-th feature (i.e. an integer value between 1 and C, assuming we have C different classes in total). In this way, in the modified version of $\boldsymbol{D}$, the distances between features belonging to the same class are preserved unchanged, while any features belonging to different classes, say class 1 and class 3, will be separated by a distance of at least $2 \times max(\boldsymbol{D})$. This will force Eqs. (1)–(3) to project such features in different sufficiently separated local neighborhoods of the resulting low–dimensional space, while at the same time the intra–class distances (the distances between features coming from the same object) are preserved. We call the low–dimensional map obtained by using Eq. (4) a "retinotopic map" (see

Fig. 1), as it resembles the topographic organization of certain structures in the brain responsive to visual input (e.g. the visual cortex, the visual nuclei in the brain stem and the lateral geniculate nucleus in the thalamus), where the centers of receptive fields of spatially adjacent neurons form an orderly sampling mosaic to cover adjacent portions of the visual field [17]. In our case, adjacent object classes are stored in adjacent areas in the low–dimensional map, without overlap.

An algorithm must also be available to map new features coming from test images. A similar problem in the context of classical MDS has been investigated in [18], where it is shown that the principal coordinates $\boldsymbol{y}_T$ for a new (unseen, or out–of–sample) data $\boldsymbol{x}_T$ are given by:

$$\boldsymbol{y}_T = \frac{1}{2}((\boldsymbol{Y}^T\boldsymbol{Y})^{-1}\boldsymbol{Y}^T)^T\boldsymbol{d}_T \tag{5}$$

$$\boldsymbol{d}^T = diag(\boldsymbol{Y}^T\boldsymbol{Y}) - \boldsymbol{d}_{T,n} \tag{6}$$

$$\boldsymbol{d}_{T,n} = (d^2(\boldsymbol{x}_T, \boldsymbol{x}_1), ..., d^2(\boldsymbol{x}_T, \boldsymbol{x}_n))^T \tag{7}$$

where $\boldsymbol{d}_{T,n}$ is a column vector of the squared distances between the test data and each of the training data. We use $\boldsymbol{d}_{T,n}$ above also to eliminate test features which are farther away than a threshold T_m from the nearest training data, as such features usually correspond to incorrectly detected keypoints. The value of T_m is determined so that to obtain best performance on the validation set of the data (explained in the next section).

Figure 1 shows a plot of the feature embedding in three–dimensional feature space constructed by sMDS (corresponding to the three largest eigenvalues, i.e. $t = 3$ in Eq. (3)). As can be seen from the figure, the features belonging to different objects (shown in different color and markers) are well separated. This is in contrast to the embedding found by PCA–SIFT (Fig. 2), where all features are intermixed and no structure is visible. For comparison, we implemented also an LDA version of SIFT (LDA–SIFT), which uses Linear Discriminant Analysis instead of PCA to map the SIFT descriptor, and the resulting map is shown in Fig. 3. As can be seen, three of the objects which happen to be quite different from the rest are separated relatively (at least partly) well, but the objects from the remaining 7 classes are all mapped to an overlapping area in the center.

4 Experimental Results

In order to demonstrate the efficacy of the proposed framework for class–specific low–dimensional feature embedding in the context of view-invariant object recognition, we have conducted several experiments in which the supervised MDS (sMDS) algorithm is compared to both PCA–SIFT and SIFT. We created a database of 10 different objects (see Fig. 4), which contains large variations in scale, viewpoint, illumination conditions and background clutter. The database is of approximately similar difficulty as the one in which [10] compares PCA–SIFT to SIFT, however our database is divided into a training, validation and

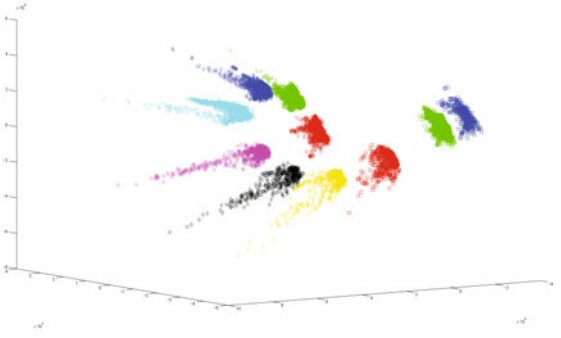


Fig. 1. 3D plot of the low-dimensional feature mapping obtained by sMDS. Different objects are shown in different color or in similar color but different markers.

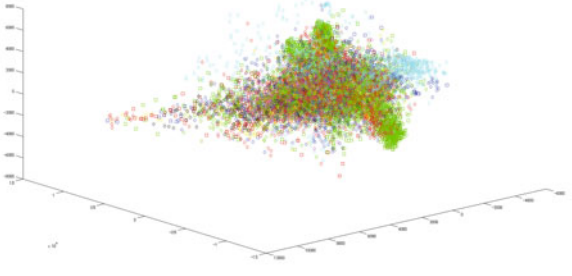


Fig. 2. 3D plot of the feature space obtained by PCA-SIFT

test sets, and has a much larger number of test images. In this way, we believe it is possible to test more accurately the performance of the different algorithms (for comparison, only 3 images per object are available in the dataset in [10]). The training set (Fig. 4a) contains 3 images for each object class, taken on predominantly black background. The validation set (see Fig. 4b) contains 5 images for each object, with a large level of background clutter, and is used to tune any parameters for each of the methods which influence its performance, e.g. the number of clusters for the bag–of–keypoints based object recognition experiment, and the matching thresholds for the feature matching based image retrieval experiment (explained below). Then, the parameters for which best performance is obtained are fixed and each method is applied to the test set, which has 10 test images for each object (see Fig. 4c). This setting simulates the common situation where a limited number of images are available for learning and validation, with a larger number of test images. Our dataset can be downloaded from `http://www.eml.hiroshima-u.ac.jp/include/database`. We have implemented the sMDS algorithm and the bag-of-keypoints algorithm in Matlab,

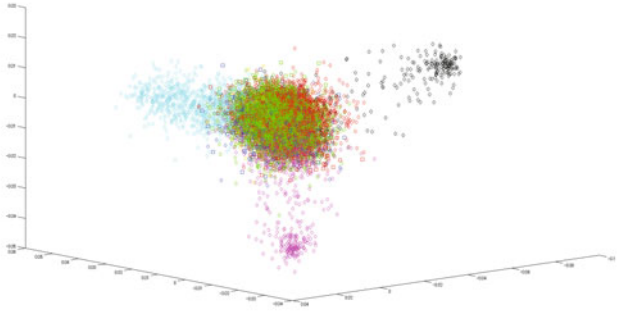
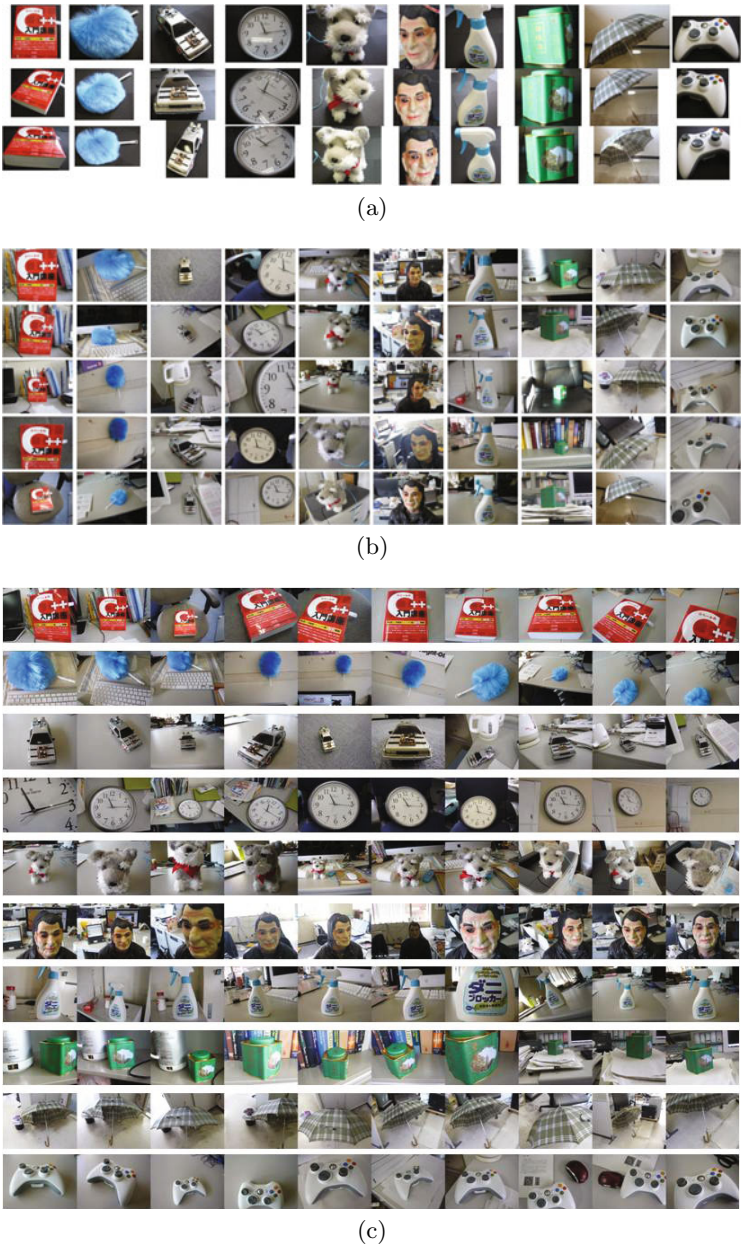


Fig. 3. 3D plot of the feature space obtained by LDA-SIFT

while for SIFT we used Lowe's original implementation and for PCA–SIFT the implementation provided by the authors of [10], which is available at [19]. In the implementation of sMDS used for the experimental evaluation, the input features $\boldsymbol{x}_i$ are identical to those used in PCA–SIFT, i.e. 3042–dimensional feature vectors obtained from the normalized gradient patches of size 41×41 pixels, centered at locations in the image where keypoints have been detected by a SIFT detector. Additionally, as a baseline, we implemented also an LDA–SIFT algorithm, where standard LDA is used instead of PCA to obtain a low-dimensional map for the features (in the case of LDA, the maximum dimension of the feature space is $C-1$, where C is the number of object classes [14]). Our implementation of the bag–of–keypoints uses the standard k–means clustering algorithm. The Euclidean distances between the histograms obtained for the test data and those obtained for the training data were compared and the class of the test images was determined by the nearest neighbor method.

The results obtained for the BoK–based object recognition experiment are shown in Fig. 5. The plots show the recognition rates (percent correctly classified test images) obtained by each method, as a function of the number of cluster centers used. As can be seen from the results, sMDS significantly outperforms the other methods. Somewhat surprisingly, the performance of PCA–SIFT is actually slightly worse than SIFT, and LDA–SIFT performs even worse than PCA–SIFT.

In the Introduction we made the conjecture (which motivated our work) that distinctive feature representation at an earlier stage might be more important than using a sophisticated classifier at a later stage. In order to test this statement, we further conducted another experiment, substituting the nearest–neighbor classifier from the previous experiment with a Support Vector Machine (SVM) using a radial basis function (RBF) kernel. The results are shown in Fig. 6. Here again sMDS significantly outperforms the other methods (note that good performance is obtained even for as few as 200 cluster centers), while the performance of the other three algorithms seems to be quite similar this time.



(a)

(b)

(c)

Fig. 4. The database used for the experiments: (a) the training set; (b) the validation set; (c) the test set

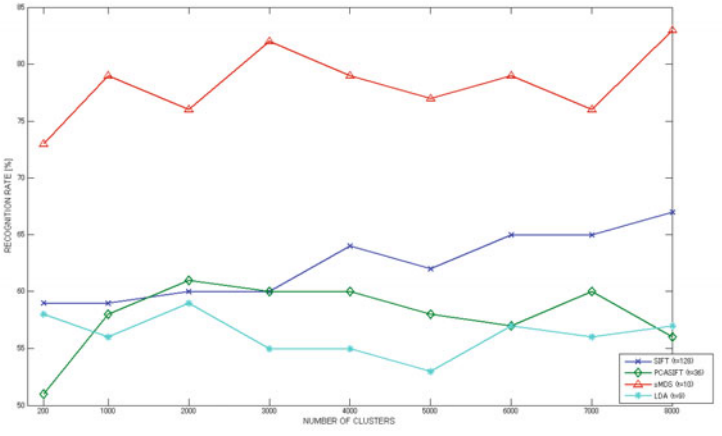


Fig. 5. Comparison of sMDS, PCA-SIFT, LDA-SIFT and SIFT for BoK (using nearest neighbor as a classifier)

Also, when SVM is used, the performance of all methods seems to be less influenced by the number of cluster centers.

Since both sMDS and PCA–SIFT construct a low–dimensional representation of the original high–dimensional features, it is interesting to see how performance depends on t, the dimension of the low–dimensional feature space. This dependence is shown in Fig. 7. For sMDS, performance seems to be stable over large range of values for t, and very good performance is achieved even for as low value as 10 dimensions. The confusion matrices obtained for each of the four methods are also given in Fig. 8 below.

In order to make sure that the improvement in recognition accuracy obtained by sMDS is valid not only for the BoK scheme, we performed another experiment, similar to the image retrieval task given in [10]. We used the same dataset as in the previous experiment, but this time recognition was based on simple feature matching. The features obtained from each test image were directly matched to the features obtained for the training images, using a threshold to determine the matching pairs. The number of matches to each class was used as a similarity measure. We used the following scoring method to evaluate the precision of the retrieval: if the correct object class was retrieved the algorithm was awarded 3 points; if the correct object appeared in the top two positions the algorithm was awarded 2 points, and if the correct object appeared in the top three positions the algorithm was awarded 1 point. Otherwise no points were given. The total scores and the correct retrieval percentage obtained for each algorithm for all test images are given in Tab. 1. The thresholds for each method were tuned to give best results on the validation set, and then the same values were used to obtain the final results on the test set. In this experiment again, sMDS achieved the best performance.

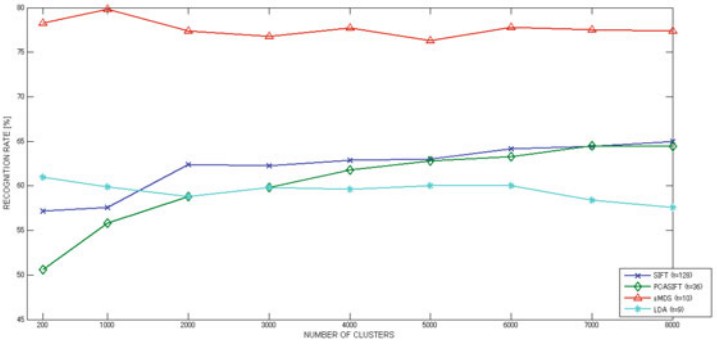


Fig. 6. Comparison of sMDS, PCA-SIFT, LDA-SIFT and SIFT for BoK (using SVM as a classifier)

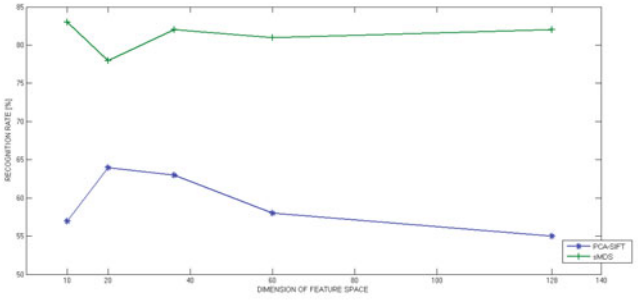


Fig. 7. Performance of sMDS and PCA-SIFT for different embedding dimensions

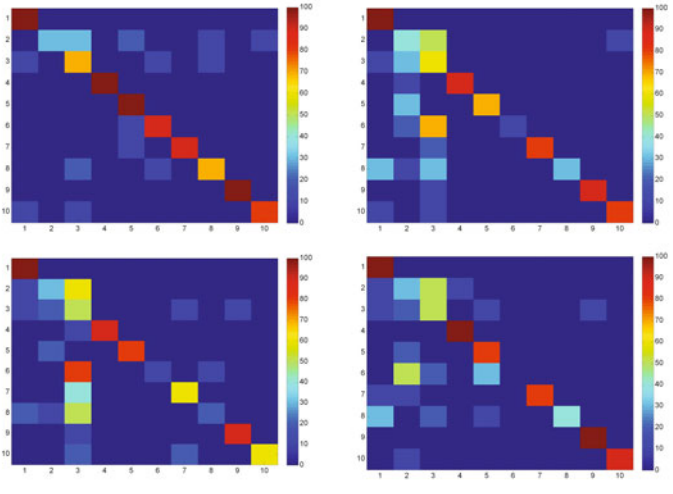


Fig. 8. Confusion matrices for sMDS (top-left), PCA-SIFT (top-right), LDA-SIFT (bottom-left) and SIFT (bottom-right)

Table 1. Results for the feature matching tasks

METHOD	FEATURE DIMENSION	THRESHOLD	SCORE	CORRECT RETRIEVAL
SIFT	128	150	125/300	41.6 [%]
PCA–SIFT	20	2600	178/300	59.3 [%]
LDA–SIFT	9	16	130/300	43.3 [%]
sMDS	10	300	221/300	73.7 [%]

5 Conclusions

In this paper we have proposed a general framework which can be used to obtain more distinctive local invariant features by projecting the original feature descriptors into low–dimensional feature space, while simultaneously incorporating class information. In the resulting feature space, the features from different classes of objects project to well-separated distinct areas. Experimental results, obtained both in a bag–of–keypoints setting and for a simple feature matching task indicate that embedding class information into the low–dimensional feature representation is beneficial and results in more accurate object recognition.

In the present implementation we have used classical MDS as a tool to obtain low–dimensional embedding of the data, because it is a well–known and theoretically well–understood method, but instead of the supervised version of MDS proposed here, a supervised version based on other linear or nonlinear dimensionality reduction methods, like manifold learning methods [20], etc. could also have been used. In a future work, we intend to compare the performance of different dimensionality reduction methods in the context of the general approach proposed in this paper. Also it would be interesting to apply the proposed method to other local feature descriptors.

References

1. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 12(60), 91–110 (2004)
2. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: SURF: Speeded Up Robust Features. Computer Vision and Image Understanding (CVIU) 110(3), 346–359 (2008)
3. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 20–25 (2005)
4. Tuytelaars, T., Mikolajczyk, K.: Local Invariant Feature Detectors: A Survey. Foundations and Trends in Computer Graphics and Vision 3(3), 177–280 (2007)
5. Murase, H., Nayar, S.: Visual Learning and Recognition of 3-D Objects from Appearance. International Journal of Computer Vision 14(1), 5–24 (1995)

6. Csurka, G., Bray, C., Dance, C., Fan, L.: Visual Categorization with bags of keypoints. In: Proc. ECCV Workshop on Statistical Learning in Computer Vision, pp. 1–22 (2004)
7. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: Proceedings of the International Conference on Computer Vision, pp. 1470–1477 (2003)
8. Cox, T., Cox, M.: Multidimensional Scaling, 2nd edn. Chapman and Hall, Boca Raton (2000)
9. Jolliffe, I.: Principal Component Analysis. Springer, Heidelberg (1986)
10. Ke, Y., Sukthankar, R.: PCA–SIFT: A More Distinctive Representation for Local Image Descriptors. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 506–513 (2004)
11. Grauman, K., Darrell, T.: The pyramid match kernel: discriminative classification with sets of image features. In: Proc. IEEE Int. Conf. Computer Vision, vol. 2, pp. 1458–1465 (2005)
12. van Gemert, J.C., Geusebroek, J.-M., Veenman, C.J., Smeulders, A.W.M.: Kernel codebooks for scene categorization. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 696–709. Springer, Heidelberg (2008)
13. Hua, G., Brown, M., Winder, S.: Discriminant embedding for local image descriptors. In: Proc. IEEE Int. Conf. Computer Vision, pp. 1–8 (2007)
14. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. IEEE Trans. PAMI 19(7), 711–720 (1997)
15. Torgeson, W.: Multidimensional Scaling: I. Theory and method. Psychometrika 17, 401–419 (1952)
16. Mardia, K., Kent, J., Bibby, J.: Multivariate Analysis. Academic Press, London (1979)
17. Wandell, B., Brewer, A.A., Dougherty, R.F.: Visual Field Map Clusters in Human Cortex. Phil. Trans. of the Royal Society London 360, 693–707 (2005)
18. Gower, J.: Adding a point to vector diagrams in multivariate analysis. Biometrica 55, 582–585 (1968)
19. http://www.cs.cmu.edu/~yke/pcasift/
20. Yan, S., Xu, D., Zhang, B., Zhang, H., Yang, Q.: Graph Embedding and Extensions: A General Framework for Dimensionality Rediction. IEEE Trans. PAMI 29(1), 40–51 (2007)

Learning Non-coplanar Scene Models by Exploring the Height Variation of Tracked Objects

Fei Yin, Dimitrios Makris, James Orwell, and Sergio A. Velastin

Digital Imaging Research Centre, Faculty of Computing,
Information Systems and Mathematics, Kingston University,
Penrhyn Road, Kingston upon Thames, Surrey,
United Kingdom KT1 2EE
{fei.yin,d.makris,james,sergio.velastin}@kingston.ac.uk

Abstract. In this paper, we present a novel method to overcome the common constraint of traditional camera calibration methods of surveillance systems where all objects move on a single coplanar ground plane. The proposed method estimates a scene model with non-coplanar planes by measuring the variation of pedestrian heights across the camera FOV in a statistical manner. More specifically, the proposed method automatically segments the scene image into plane regions, estimates a relative depth and estimates the altitude for each image pixel, thus building up a 3D structure with multiple non-coplanar planes. By being able to estimate the non-coplanar planes, the method can extend the applicability of 3D (single or multiple camera) tracking algorithms to a range of environments where objects (pedestrians and/or vehicles) can move on multiple non-coplanar planes (e.g. multiple levels, overpasses and stairs).

Keywords: Camera calibration, non-coplanar planes, region segmentation, motion variety, depth and altitude estimation.

1 Introduction

In recent years, a significant amount of research effort has been put on 3D pedestrian tracking from single or multiple surveillance cameras. Most of the existing methods that perform 3D object tracking, assume that all objects move on a single flat ground plane that is defined either manually or automatically from tracking observations. However, such a simple model is not able to handle scenes that contain multiple non-coplanar structures such as ramps, stairs and overpasses. In this paper, we propose a method for estimating the 3D geometry for such scenes from noisy 2D observations of walking pedestrians.

Hoiem *et al* [1] proposed a probabilistic modeling of the scale and location variance of objects in the scene, thus they built up a relationship between the size of objects and their positions and then could filter out false detections. Saxena *et al* [2] assumed that the world consists of vertical structures and a single flat ground surface. Then, a classifier was trained to model the relation between local

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 262–275, 2011.

material properties (colour and texture), 3D orientation, and image location. Other automatic ways for calibrating a ground plane from observed tracks of walking people were proposed in [3], [4] and [5], which assumed accurate head and foot positions of single pedestrians. All the above methods can only deal with situations where all the objects move on a single coplanar ground plane. Breitenstein *et al* [6] proposed an online learning approach for estimating a rough 3D scene structure from the outputs of a pedestrian detector. They divide the image into small cells and compute the relative depth for each image cell. However, their scene model is actually a depth map that does not explicitly represent the real 3D spatial dimensions of scene features.

Different from other methods, the main novelty of the proposed approach is the accumulation of evidence for the presence of different planar regions in the scene through pedestrian tracking, once enough tracks are available, a form of clustering is applied and each image pixel is associated with a cluster which defines a separate planar surface in the scene. The framework for estimating non-coplanar planes is summarized in the following diagram:

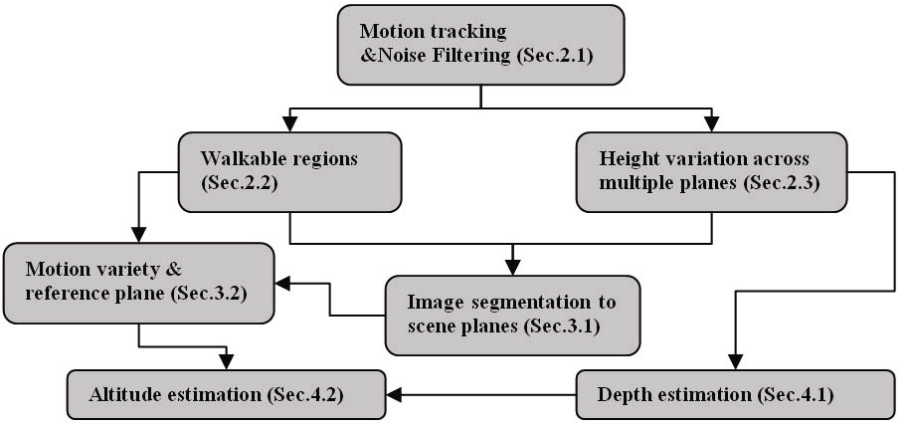


Fig. 1. Framework overview

1.1 Camera Projection Model

In this work, we use a model which assumes a linear relationship between the 2D image height of an object and its image vertical position (see Fig. 2), similar to [7]:

$$h = R(y_B - H_L) \tag{1}$$

where h is the object 2D image height, R is the object height expansion rate, y_B is the vertical image position of the detected object (foot position) and H_L is the image y-coordinate of the horizon line. The object pixel height h is zero at the horizon H_L and maximum at the bottom row of the image. Note that this projection model can only apply for objects moving on a single flat plane.

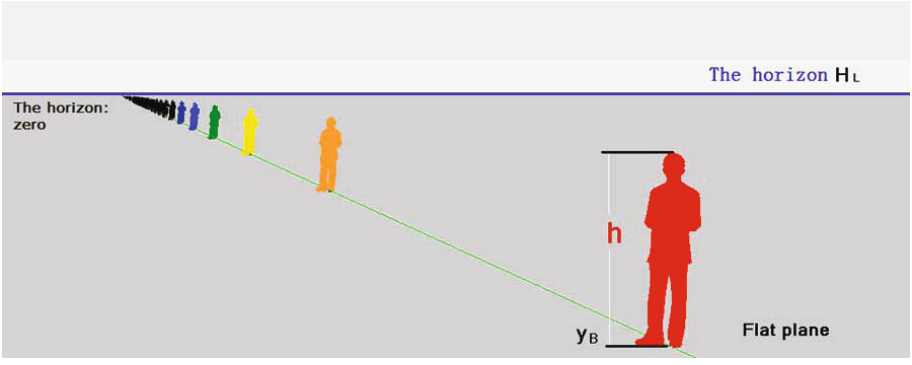


Fig. 2. Camera projection model

This camera projection model assumes that the camera roll angle is zero so the horizon is parallel to the x-axis. When this is not the case, an image transformation can be applied to satisfy this condition. In addition, the rest of camera parameters (e.g. tilt angle, height, focal length) have appropriate values that allow the variation of objects sizes with respect to their y coordinates. The above assumptions are typical for the majority of surveillance cameras.

1.2 Image Patch Model

The image plane is divided uniformly into small patches $P_{m,n}$, where m and n are the row and column index of each patch:

$$P_{m,n} = \{W_{m,n}, \mu^H_{m,n}, A_{m,n}, (c_{m,n}, d_{m,n})\} \tag{2}$$

where $W_{m,n}$ is a binary variable that indicates whether this image patch is walkable or not (Sec.2.2), $\mu^H_{m,n}$ is the average pedestrian height located in this patch (Sec.4.1), $A_{m,n}$ is the estimated altitude (Sec.4.2), and *($c_{m,n}$, $d_{m,n}$)* are line parameters that indicate the relationship between pedestrian height and image vertical positions (Sec.3.1).

2 Processing of Tracking Observations

2.1 Motion Tracking

For each pedestrian in the scene, a track (or an observation) is derived by a blob tracking algorithm, e.g. [13]. For a pedestrian j *= [1,2..M]*, a track O_j is defined as:

$$O_j = \{[x^{min}_{j,k}, x^{max}_{j,k}, y^{min}_{j,k}, y^{max}_{j,k}]\} \tag{3}$$

where k is the frame number. The bounding box $[x^{min}_{j,k}, x^{max}_{j,k}, y^{min}_{j,k}, y^{max}_{j,k}]$ defines the object width ($W_{j,k} = x^{max}_{j,k}$ - $x^{min}_{j,k}$) and height ($H_{j,k} = y^{max}_{j,k}$ - $y^{min}_{j,k}$) and its centre bottom point ($B_{j,k}$,$C_{j,k}$). where $B_{j,k}$ is the lower y-coordinate of the

bounding box ($B_{j,k} = y_{j,k}^{max}$) and $C_{j,k}$ is the middle x-coordinate ($C_{j,k} = (x_{j,k}^{min} + x_{j,k}^{max})/2$).In practice, before any further processing, it pays to filter tracks to remove unreliable measurements. In our case, we use the LOWESS method [14] to smooth the bounding box sizes for each track (10% of the track length as the window size), and also remove bounding boxes where the ratio between the height and width is below a threshold T_{hw} (experimentally set to 2), which are likely to violate the assumption that we are processing walking pedestrians only.

2.2 Walkable Regions

For a given scene, people normally appear on regions which can be called "walkable" (e.g. not on walls or buildings). Therefore, detecting where people appear can help us to identify walkable regions in the camera FOV. A patch is walkable if the number of observations ($B_{j,k}$,$C_{j,k}$) located inside a patch is above a threshold T_{hw}. Then, by a connected component analysis, image patches are grouped and labeled as walkable regions (see Fig.11). Walkable regions will be further segmented in Sec.3.1.

2.3 Height Variation across Multiple Planes

The linear camera projection model (Eq.1) is valid if objects move on a single flat plane. However, this is not true for scenes that contain ramps or stairs. Fig.3 shows that when a pedestrian moves across different planes (at the boundary between the flat area and the stairs at around y=480), there is a noticeable change of slope of the object height/y-axis plot.

We adopt a Hough transform approach to detect the slope change and consequently determine the number of planes for each walkable region. Let's assume that the frame span of a track O_j is from $K_{j,o}$ to $K_{j,p}$. Firstly, we divide the track uniformly in time into N parts. Each track segment i (i = [1...N]), consists of a set of points Q_i=($B_{j,k}$,$H_{j,k}$), where k is the frame index of Q_i, ($K_{j,p}$-$K_{j,o}$)/N is

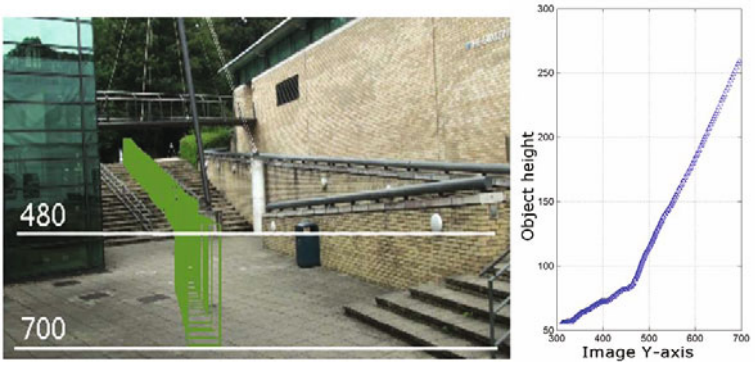


Fig. 3. Bounding boxes of a tracked pedestrian j (left) and the relationship between object heights $H_{j,k}$ and vertical position on the image of $B_{j,k}$ (right)

the length of each track segment.Each point ($B_{j,k}$,$H_{j,k}$) reflects the relationship between pedestrian heights and the vertical position on the image plane. Then, we perform least square line fitting for all points between $K_{j,o}$ and $K_{j,p}$. We find the line parameters ($c_{j,i}$,$d_{j,i}$) in slope-intercept form, which minimize the average square distance from points to the line segment.

The i^{th} fitted line function for track j is:

$$H_{j,k} = c_{j,i}B_{j,k} + d_{j,i} \tag{4}$$

and the average square distance error is:

$$E = \sum_{k} \frac{(H_{j,k} - c_{j,i}B_{j,k} - d_{j,i})}{(K_{j,p} - K_{j,o})/N} \tag{5}$$

Therefore, for each track O_j, we obtain a set of line parameters ($c_{j,i}$,$d_{j,i}$) or equivalently ($\theta_{j,i}$,$S_{j,i}$), where $\theta_{j,i} = arctan(c_{j,i})$ is the angle between each line and the x-axis and $S_{j,i} = -d_{j,i}/c_{j,i}$ is the intercept. Each fitted line represents a linear relationship between the pedestrian height and the image vertical position or equivalently a plane that the pedestrian moves on. For further analysis, a histogram of angles $\theta_{j,i}$ is obtained. Fig.5 shows that for pedestrians moving across planes, their height curves will change its slope and more than one peak will occur in the histogram (left side of Fig.5). For pedestrians moving only on one of the planes (right side of Fig.5), their height curves will be a single line ideally, the variation of angles of fitted lines will be small and one peak will occur in the histogram.

Finally, we obtain the histogram of angles of all the tracks for a specific walkable region. After applying a moving average to smooth the histogram, we obtain all the peaks (local maxima). Each peak corresponds to a plane in the scene and is described as a single Gaussian:

$$(\mu_i^{\theta}, \sigma_i^{\theta}, \mu_i^{S}, \sigma_i^{S}) \; i = 1...N_{class} \tag{6}$$

where μ_i^{θ} , σ_i^{θ} are the mean and standard deviation of angle $\theta_{j,i}$, and μ_i^{S}, σ_i^{S} are the mean and standard deviation of intercepts $S_{j,i}$ for each class i. N_{class} is the total number of classes for the given walkable region (see Fig.12).

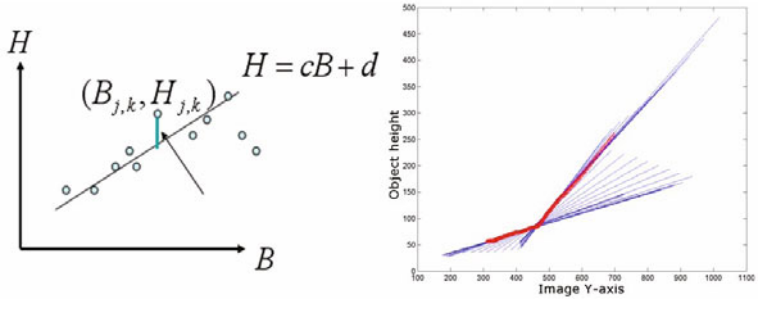


Fig. 4. Least square line fitting (red: tracking data points, blue: fitted lines)

3 Image Segmentation to Scene Planes

3.1 Segmentation of Walkable Region

After the number of planes (classes) for a given walkable region were estimated as described in the previous section, all image patches are classified to different planes. The steps to segment a walkable region into planes are summarized as follows:

1. For each image patch $P_{m,n}$ of the walkable region, we obtain all the tracked pedestrians ($B_{j,k}$,$H_{j,k}$) whose centre bottom points are located inside this patch (see Fig.6).

2. A least square line fitting algorithm is applied to obtain the line parameters $c_{m,n}$,$d_{m,n}$ for this image patch. The angle between the line and the x-axis, $\theta_{m,n}$= *arctan($c_{m,n}$)* and the intercept $S_{m,n}$= -$d_{m,n}/c_{m,n}$, will then be used as a feature of this image patch in order to classify it into different planes.

3. A segmentation method similar to the one described in [9] is applied. The image patch $P_{m,n}$ is labelled by the class (plane) i (Eq.6) that minimizes the difference:

$$Arg \min_{i \in [1, N_{class}]} [\alpha \frac{(\theta_{m,n} - \mu_i^{\theta})^2}{(\sigma_i^{\theta})^2} + (1 - \alpha) \frac{(S_{m,n} - \mu_i^{S})^2}{(\sigma_i^{S})^2}, i] \quad (7)$$

where $\theta_{m,n}$ is the angle feature for the image patch, and $S_{m,n}$ is the intercept for the image patch, and α controls the combination weights between the two parts.

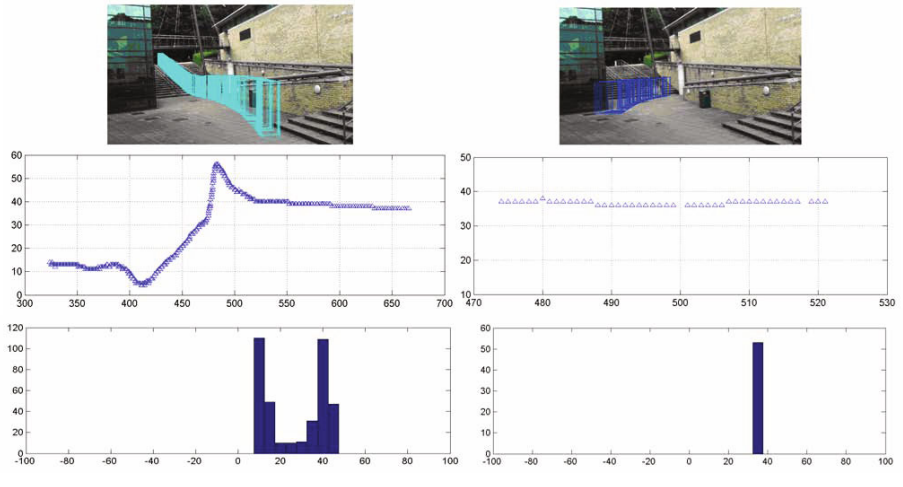


Fig. 5. Example of line angles and their histograms (Top: bounding boxes of pedestrians, middle: angles, bottom: histogram of angles)

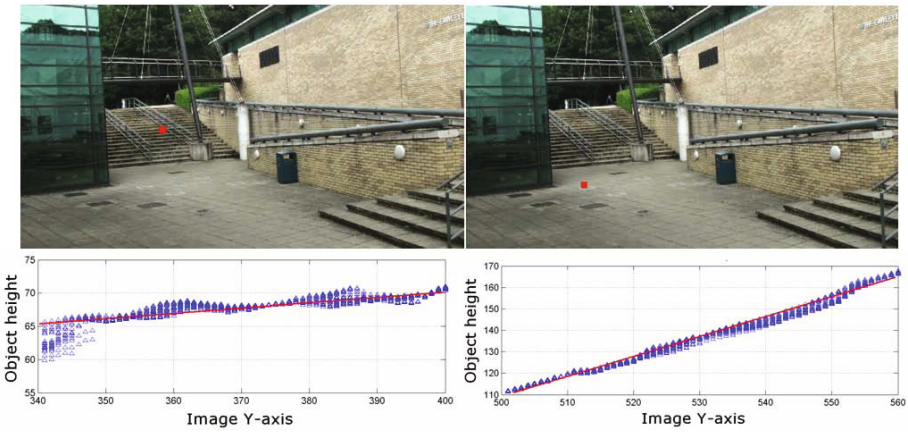


Fig. 6. Examples of line features for image patches (the red rectangles)

4. Due to noise, a few image patches get an incorrect label during step 3. To address this issue, the label of an image patch may change by minimizing the following cost function:

$$Arg \min_{i \in [1, N_{class}]} [\frac{(\theta_{m,n} - \mu_i^{\theta})^2}{(\sigma_i^{\theta})^2} + \beta \sum_{o=m-c,k=n-c}^{m+c,n+c} \frac{\mu_{o,k}}{|\theta_{m,n} - \theta_{o,k}|}] \tag{8}$$

the cost function takes the difference between the patch $P_{m,n}$ and its neighbour patches into consideration (assuming eight neighbours here). $\mu_{o,k}$=*0* when $P_{m,n}$ and $P_{o,k}$ have the same label, and $\mu_{o,k}$=*1* when $P_{m,n}$ and $P_{o,k}$ have different label. The parameter β is set experimentally to 0.5

5. We repeat step 4 until no change of class label is observed.

3.2 Global Motion Variety

People are likely to move towards certain directions when they move on certain geometric structures. For example, people often follow the path on a bridge, also, go straight up or down on stairs statistically. Since their motion patterns can differ on different planes (e.g. the overpass, stairs, the ground), we detect such difference to distinguish between planes and define a reference plane. We compute statistics about which direction each pedestrian takes and how many times: each time a pedestrian's centre bottom point $B_{j,k}$ is located within $P_{m,n}$, we compute a motion vector for the next few frames which indicates the direction of the pedestrian's motion., Then, all motion directions are accumulated to a histogram of motion directions, consisting of N_v direction bins, as shown in Fig.7.

$$\{V_i\} \; i = [1...N_v] \tag{9}$$

where i indicates the direction of motion (N_v= *4* in this work), and V_i is the count of the number of times pedestrians have taken that direction.

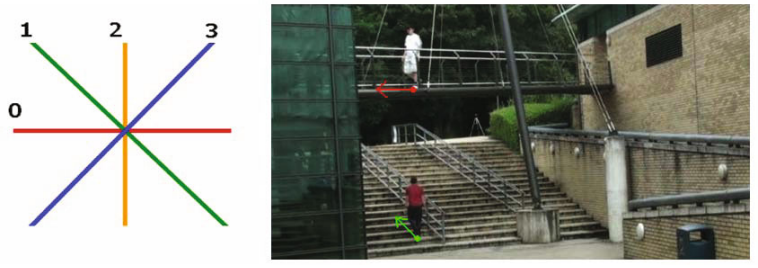


Fig. 7. Four direction motion mode

We compute the "motion variety" for each image patch $P_{m,n}$ as follows:

$$V_{m,n} = \{\frac{V_i}{\sum_{i=1}^{N_v} V_i}, \sqrt{\frac{\sum_{i=1}^{N_v}[V_i - max(V_i)]^2}{N_v - 1}}\} \tag{10}$$

Then, a reference plane needs to be chosen arbitrarily and the rest of the planes are defined relative to this reference plane. We choose the region with largest motion variety as the reference plane. Although this reference plane is not necessarily the flat ground plane, it is more likely to be a plane parallel to the ground plane, as stairs and slopes tend to have smaller motion variety (see Fig.14b).

4 3D Scene Model Estimation

4.1 Estimating Average Heights

A relative depth map is established by accumulating height observations of tracked objects for each image patch. We model the noise observations of object heights of each patch with a single Gaussian to address the issue of noisy. Specifically, for each patch $P_{m,n}$, the pedestrian height $H_{j,k}$ information is obtained, whose ($B_{j,k}$,$C_{j,k}$) is located inside this patch. Then, the heights are modelled as a mean $\mu^H_{m,n}$ and standard deviation $\sigma^H_{m,n}$. Note that some image patches will have very few or no observation at all which implies that those areas are not walkable by pedestrians.

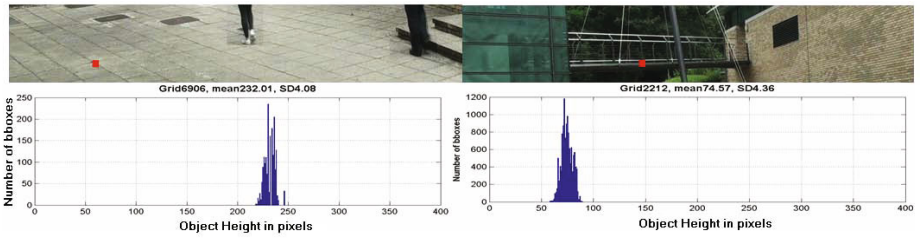


Fig. 8. Pedestrian height information for image patches (red rectangles)

4.2 Altitude Estimation

As the reference plane has been chosen in section 3.2 and the pedestrian height information for each image patch has also been obtained in section 4.1, the next step is to estimate the relative altitude for each image patch in the scene with regard to the reference plane.

As illustrated in Fig.8, for each image patch (red rectangles), an average pedestrian height $\mu^H_{m,n}$ is obtained as mentioned in section 4.1. One can always find a position with the same pedestrian height somewhere on the reference plane using Eq.11. $y^r_{m,n}$ is called the reference vertical position (green rectangles).

$$y^r_{m,n} = \frac{\mu^H_{m,n}}{R_r} + y_h \tag{11}$$

The expansion rate R_r and the horizon y_h (where the pedestrian height is zero) for the reference plane is estimated using the line fitting method mentioned in section 3.1.

If there is a difference between the vertical position of the image patch and the reference vertical position $y^r_{m,n}$, this indicates that the image patch may not be located on the reference plane but on other planes that are higher or lower than the reference plane. We estimate the relative altitude $Ar_{m,n}$ for the image patch $P_{m,n}$ by taking the difference of vertical positions and normalize it by the average pedestrian's height $\mu^H_{m,n}$, we will get an estimation of the relative altitude $Ar_{m,n}$ for the image patch:

$$Ar_{m,n} = \frac{(y_{m,n} - y^r_{m,n})}{\mu^H_{m,n}} \tag{12}$$

Finally, we assume an average height of pedestrians H_{av} (e.g. 1.70 meters) to convert the altitude of each image patch $P_{m,n}$ into meters.

$$A_{m,n} = Ar_{m,n} H_{av} \tag{13}$$

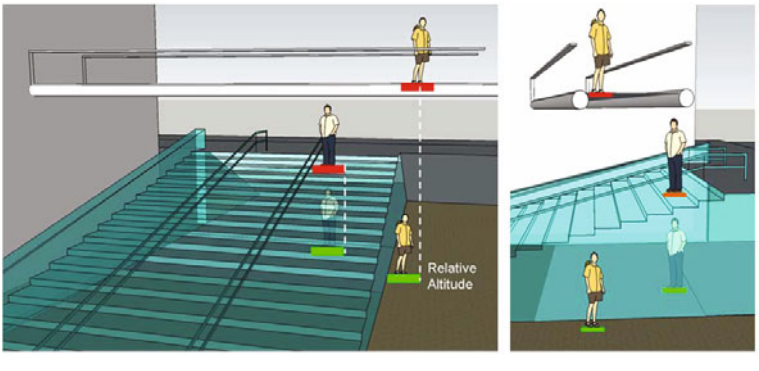


Fig. 9. Illustration of how altitude been estimated

5 Dataset and Results

5.1 Dataset

The dataset used in this work is called Kingston Hill dataset which is captured in Kingston Hill campus of Kingston University, London and is available at http://dipersec.kingston.ac.uk/NCGMdata. It is a multiple camera dataset with two cameras monitoring roughly the same area and time synchronized. These videos were recorded by HD cameras. The image resolution is 1280×720. The dataset contains several hours of videos with pedestrians moving around frequently (with low object density in the scene). There are non-coplanar structures in the scene such as stairs and overpass.

To the best of our knowledge, there is no existing public surveillance dataset which deals specifically with scenes of multiple non-coplanar planes. Therefore, we have made our dataset public available to allow researchers to work on tracking in multi-planar environments and compare results.

5.2 Results and Evaluation

In order to verify our method, we tested our algorithm on the dataset described above. The frames from HD videos are divided into regular 10pix×10pix patches. A motion tracker is used to obtain the position and size of each pedestrian when they walk through the scene and more than 200 tracks are obtained. In Fig.11, we show the results of grouping the camera FOV into walkable regions (Sec.2.2).

Fig.12a and Fig.12b show the result of the histogram of angles of all the tracks for each walkable region (Sec.2.3). Fig.13a and Fig.13b show the intermediate and final scene segmentation results respectively (Sec.3.1). At this stage, the walkable region (1) was split to a flat area (red) and the stairs (green).

Fig.14a shows the motion variety for different coplanar regions for camera one (Sec.3.2). We can see that the motion vectors on the overpass are very clear and uniformed (mainly on direction 0). Motion on the stairs is fairly uniformed (mainly on direction 1). However, on the flat area, the motion vectors are diverse, therefore, the motion variety will be larger.

Fig.14b shows a relative depth map based on average pixel-wise pedestrian height for each image patch where different colours represent different pedestrian heights (Sec.4.1).

Fig. 10. Kingston Hill Dataset, camera view 1(left) and camera view 2(right)

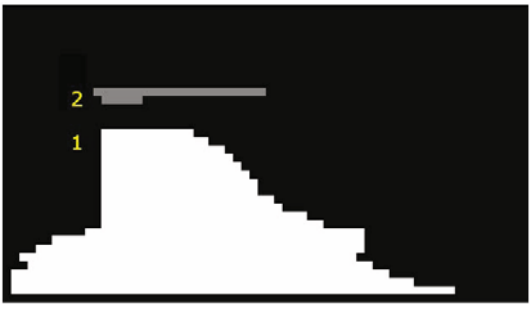


Fig. 11. Group of walkable regions for camera1

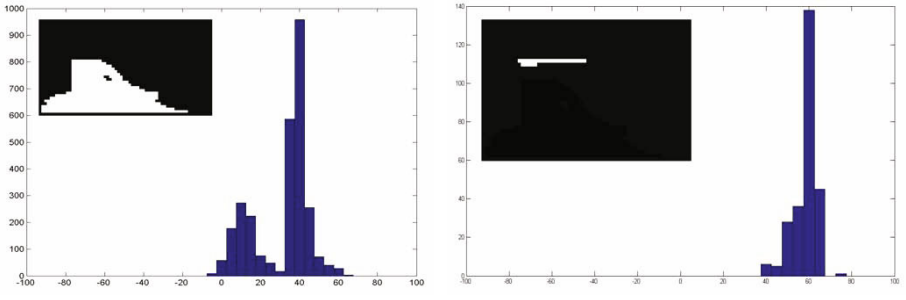


Fig. 12. a) Histogram of angles of lines for walkable region1 b) Histogram of angles of lines for walkable region2

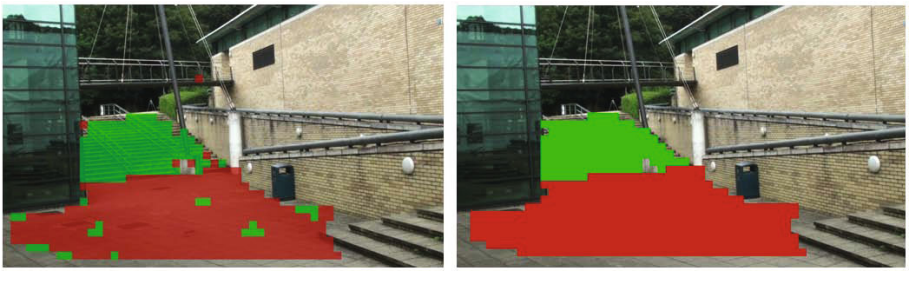


Fig. 13. a) Intermediate segmentation result for camera 1, b) Final segmentation result for camera 1

Fig.15a and Fig.15b show the results of estimated altitude for each image patch of both camera views. The x, y axes are the image coordinates and the z axis is the estimated altitude. We can see a rough 3D structure of the scene: the flat area, the stairs and the overpass which is higher than the stairs. In order to evaluate the accuracy of the altitude estimation, we measured the real sizes of the stairs and overpass. There are 19 steps, the first 18 of them are 18 cm in

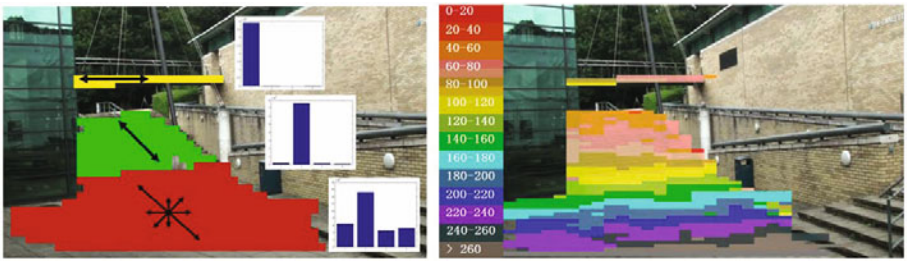


Fig. 14. a) Global motion variety for camera1 with histograms showing motion direction frequency, b) Pedestrian height for each image patch for camera1

Table 1. Evaluation on altitude estimation in meters

	Ground truth	Camera1	Camera2
Overpass:	5.0	5.1	5.0
Stairs:	3.4	3.3	3.5
Flat area:	0.0	0.0	0.1

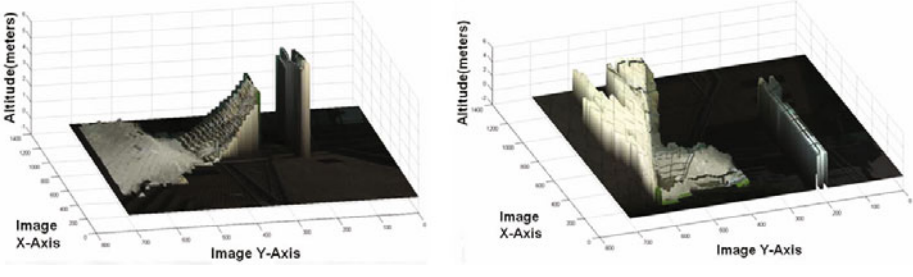


Fig. 15. a) Estimated attitude for each image patch for camera1, b) Estimated attitude for each image patch for camera2

height, and the last step is 16 cm in height. Hence the height of the stairway is 3.4 meters in total. The height of the overpass is 5 meters.

In table 1, we can see that our estimations of the heights of the stairs and the overpass for the first camera view are 3.3 and 5.1 meters respectively (3.5 and 5.0 for the second camera view). Therefore, the proposed method estimated accurately (overall error: less than 0.1 meter) the real altitude for 3D scene structures.

6 Conclusion and Future Work

We proposed a method to automatically estimate a non-coplanar scene model by statistically exploring the variation of pedestrian heights across the camera FOV. The proposed method is able to find out the relative depth, segment the image plane into regions which belong to the same geometric coplanar plane, identify a reference plane and estimate the altitude for each image pixel, thus building up a 3D scene model which contains multiple non-coplanar planes. Such a method is very useful for surveillance applications, as it allows 3D (single or multiple camera) tracking in scenes which contain non-coplanar structures such as multiple levels, overpasses and stairs. We also demonstrated that our estimation of altitude is sufficiently accurate.

For future work, we aim to build up a 3D model which reflects the real world scale of the scene structures by taking the real scale of pedestrians into consideration. We also aim to extend our method to multiple cameras. Such an approach will not only produce more accurate 3D representations but also will map each of the 2D image views into a common 3D scene model that will allow multiple-camera tracking in wide area non-coplanar environments.

References

1. Hoiem, D., Efros, A.A., Hebert, M.: Putting objects in perspective. International Journal of Computer Vision, 0920–5691 (2008)
2. Saxena, A., Sun, M., Ng, A.: Make3D: Learning 3D Scene Structure from a Single Still Image. IEEE Trans. on Pattern Analysis and Machine Intelligence, 824–840 (2009)
3. Renno J.R., Orwell J., Jones G.A.: Learning Surveillance Tracking Models for the Self-Calibrated Ground Plane. In: British Machine Vision Conference, Cardiff, pp. 607–616 (September 2002)
4. Krahnstoever, N., Mendonca P.: Autocalibration from Tracks of Walking People. In: British Machine Vision Conference, Edinburgh, UK (2006)
5. Lv, F., Zhao, T., Nevatia, R.: Camera calibration from video of a walking human. IEEE Trans. on Pattern Analysis and Machine Intelligence, 1513–1518 (2006)
6. Breitenstein, M.D., Sommerlade, E., Leibe, B., van Gool, L., Reid, I.: Probabilistic Parameter Selection for Learning Scene Structure from Video. In: British Machine Vision Conference (BMVC 2008) (2008)
7. Greenhill, D., Renno, J.R., Orwell, J., Jones, G.A.: Occlusion Analysis: Learning and Utilising Depth Maps in Object Tracking. In: Image and Vision Computing, Special Issue on the 15th Annual British Machine Vision Conference, vol. 26(3), pp. 430–444. Elsevier Publishing, Amsterdam (2008)
8. Huang, C., Wu, B., Nevatia, R.: Robust object tracking by hierarchical association of detection responses. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 788–801. Springer, Heidelberg (2008)
9. Lin, L., Zhu, L., Yang, F., Jiang, T.: A novel pixon-representation for image segmentation based on Markov random field. Image and Vision Computing 26(11), 1507–1514 (2008)
10. Loy, C.C., Xiang, T., Gong, S.G.: Modelling Multi-object Activity by Gaussian Processes. In: British Machine Vision Conference (BMVC 2009) (2009)

11. Madden, C., Cheng, E.D., Piccard, M.: Tracking people across disjoint camera views by an illumination-tolerant appearance representation. Machine Vision Applications 18(3-4), 233–247 (2007)
12. Fernandes, L.A.F., Oliveira, M.M.: Real-time line detection through an improved hough transform voting scheme. Pattern Recognition 41(9), 299–314 (2008)
13. Xu, M., Ellis, T.J.: Partial observation vs. blind tracking through occlusion. In: British Machine Vision Conference, BMVA, Cardiff, pp. 777–786 (September 2002)
14. Cleveland, W.S., Devlin, S.J.: Locally-Weighted Regression: An Approach to Regression Analysis by Local Fitting. Journal of the American Statistical Association 83(403), 596–610 (1988)

Optimal Regions for Linear Model-Based 3D Face Reconstruction

Michaël De Smet and Luc Van Gool

K.U. Leuven ESAT-PSI/VISICS, Heverlee, Belgium
michael.desmet@esat.kuleuven.be

Abstract. In this paper, we explore region-based 3D representations of the human face. We begin by noting that although they serve as a key ingredient in many state-of-the-art 3D face reconstruction algorithms, very little research has gone into devising strategies for optimally designing them. In fact, the great majority of such models encountered in the literature is based on manual segmentations of the face into subregions. We propose algorithms that are capable of automatically finding the optimal subdivision given a training set and the number of desired regions. The generality of the segmentation approach is demonstrated on examples from the TOSCA database, and a cross-validation experiment on facial data shows that part-based models designed using the proposed algorithms are capable of outperforming alternative segmentations w.r.t. reconstruction accuracy.

1 Introduction

Many problems in computer vision deal with objects that can be subdivided into meaningful parts by a human observer. It is widely believed—both in psychology [1] and in computer science—that such a decomposition can enhance our understanding of an object. This may enable us for example to identify partially occluded or locally deformed objects, or to extrapolate from known examples of an object class. Because of its social relevance, one of the most frequently studied object classes in computer vision is the human face.

In this paper we demonstrate that—taking faces as a good case in point—the optimal subdivision into parts does not follow the intuitive subdivisions that have been used so far. We derive a method to extract better parts and show their superiority in 3D face reconstruction experiments.

Many authors have demonstrated the usefulness of intuitive part-based representations in automatic face recognition tasks. In one of the earliest works, Brunelli and Poggio [2] showed that a template matching scheme based on a combination of facial features such as the eyes, nose, and mouth provides better facial recognition rates than a similar technique based on the face as a whole. More recently, variations on this approach incorporating eigenfeatures have proven to be particularly useful when dealing with partial occlusions and facial expressions [3,4,5]. Similar results have also been found for 3D face recognition [6].

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 276–289, 2011.

An important aspect of part-based representations is that they enable more accurate reconstructions of novel examples of the object class. Blanz and Vetter [7] augmented their 3D Morphable Model (3DMM) of the human face by manually partitioning the face into four regions. By independently adjusting the shape and texture parameters for these regions, and blending the results into a single face model, they were able to obtain more accurate 3D face reconstructions than with a holistic approach. Similarly, Peyras et al. [8] used region-specific Active Appearance Models (AAMs) to enable accurate facial feature fitting on unseen faces. The same principle has been adopted by various authors [9,10,11,12,13,14] to enhance the performance of 3DMMs in 3D face modeling, 3D face reconstruction, and automatic face recognition tasks. The main difference between these contributions regarding the part-based representation lies in the way the parts are joined at the boundaries to form a complete face model.

The previously mentioned works have abundantly shown the merits of part-based representations, but they do not provide any automatic tools for obtaining the subdivision into parts, instead relying on manual segmentation of the regions. While this approach may be acceptable for objects where the underlying regions are intuitively clear, other object classes may benefit from automatic partitioning techniques. Furthermore, we will demonstrate that even for familiar object classes like the human face, a manual segmentation is not necessarily optimal.

In the literature, a large amount of research has gone into the development of automatic 3D mesh segmentation techniques [15]. The vast majority of these are based on geometric properties such as curvature or geodesic distances. While these methods tend to work well for articulated objects like full-body scans, they are less reliable for faces, where the parts are ill-defined from a geometrical standpoint. Indeed, we believe that in general a method for automatically subdividing an object class into meaningful parts should not be based solely on geometric properties, and could benefit greatly from deformation statistics. This is especially true when the available data is not geometrical in nature, which for example is the case for color images. In this paper, we demonstrate that good segmentations can be obtained using only deformation statistics.

For automatic blendshape segmentation in facial animation, Joshi et al. [16] proposed to apply a thresholding operation to a maximum deformation map, followed by some post-processing to clean up the resulting regions. A promising candidate for automatic object decomposition is the Nonnegative Matrix Factorization (NMF) framework, due to Lee and Seung [17]. NMF and its relatives [18,19] have been applied to databases of facial images, resulting in a set of nonnegative basis images capable of reconstructing the original images with minimal error. By applying sparseness constraints, the basis images can be made to correspond more or less with facial features, but it is not entirely clear how to extract distinct facial regions.

While researching transform invariant models for pedestrian detection, Stauffer and Grimson [20] introduced the concept of Similarity Templates as a statistical model of pixel co-occurrences within images of the same object class. By applying hierarchical clustering to an aggregate Similarity Template of a database

of aligned pedestrian images, they were able to automatically construct a region segmentation that corresponds well to meaningful parts of pedestrian images. Our approach is similar as it uses statistical information about the relationships between vertices in an aligned database of 3D face models, and applies clustering techniques to obtain a decomposition into regions of high correlation. Additionally, we present a technique to automatically determine optimal blending weights for recombining facial parts into a complete 3D face model.

2 Preliminaries

Before introducing our methods for automatically subdividing an object, it is worth mentioning the problem we set out to solve. Although the data used in our experiments was derived from a set of laser scanned faces, the formulation and algorithms are general enough to be applied to any object class that can be modeled as a linear combination of eigenfeatures.

2.1 Facial Data

The facial data used in this paper is based on the USF DARPA HumanID 3D Face Database of laser scanned faces in a neutral pose. A subset of 187 laser scans has been selected from the original database so that each person's face is present only once in the dataset. The laser scans have been brought into dense correspondence using a regularized non-rigid registration procedure derived from [21]. The resulting dataset consists of 187 3D face shapes, each composed of 60 436 vertices. Encouraged by the results of [22,23,24], we decided to exploit the mirror symmetry properties of the human face space by extending the dataset with a mirror image of each face.

2.2 Linear Subspaces for Reconstruction

Given a set of data vectors, the optimal set of basis vectors for linearly reconstructing the original set in the least squares sense is given by Principal Component Analysis (PCA). This inherently entails some restrictions.

1. The reconstructions are linear combinations of basis vectors. Better results may be possible when this linearity restriction is removed.
2. The reconstruction error is minimized in a least squares sense. Depending on the application, this may not be the best error measure.
3. Minimal squared reconstruction error is only guaranteed when reconstructing vectors from the training set. When a dataset is split into a training set and a test set, the basis vectors obtained by applying PCA to the training set may not optimally reconstruct the vectors in the test set.

The third issue is what we are trying to address in this paper. It occurs when the training set is not large enough for PCA to reliably estimate all the modes of variation in the population, which is normally always the case when dealing

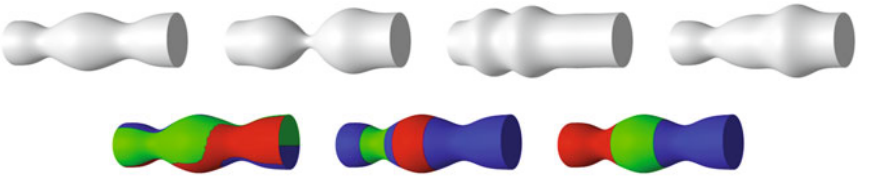


Fig. 1. Example of applying the segmentation technique to synthetic data. The data was generated by applying three independent overlapping gaussian deformations to a cylinder. *Top row:* four of the 20 available training examples. *Bottom row:* three-component segmentations based on (*left*) statistically normalized displacement vectors, (*center*) displacement magnitudes, (*right*) statistically normalized displacement magnitudes. Only the last type of feature vectors leads to the correct part subdivision.

with non-synthetic data. One way of improving the reconstruction quality for vectors outside the training set is by incorporating prior knowledge about certain regularities in the population. For example, the mirror symmetry of human faces can be exploited by adding mirrored examples of the original faces to a training set. When the number of training vectors is less than the size of the vectors, this trivially boosts the reconstruction quality by increasing the number of degrees of freedom in the PCA model. Much more importantly, as shown in [23] for grayscale images of faces, this also improves the quality of the computed basis vectors, resulting in increased signal-to-noise ratios for reconstructions even when using a fixed number of basis vectors. This effect was shown to persist even with large training sets of 5627 images of 64×60 pixels. Another popular approach is to subdivide the original training vectors into separate regions, preferably corresponding to localized features in the object class, and train a PCA model on each of those regions. This is known as the *eigenfeatures* approach [3]. By doing so, the number of available basis vectors is multiplied by the number of subregions, resulting in greater representational power, while retaining the ability to perfectly reconstruct the original training vectors. In principle, one could keep subdividing the training vectors into more and more regions until the desired reconstruction accuracy is achieved. However, in most applications it is desirable to keep the number of basis vectors as low as possible. Therefore, our objective is—given a limited number of regions—to automatically find those regions that minimize the reconstruction error outside the training set. It is expected that these regions will correspond to meaningful parts of the object class.

3 Automatic Segmentation of Facial Regions

Suppose we have a training database of M objects, each sampled at N corresponding vertex locations. When subdividing this set of 3D surfaces into regions, we wish to cluster vertices together according to some measure of similarity. In this section, we will design a similarity measure suitable for this context.

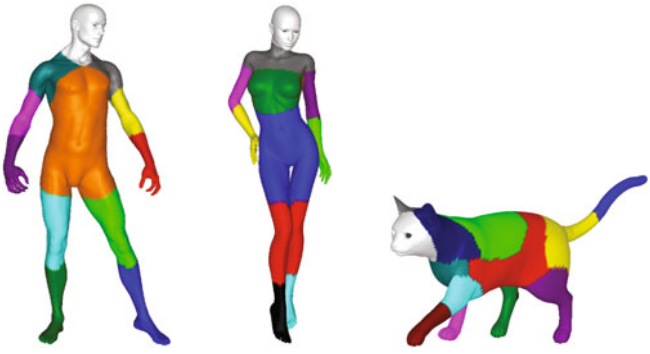


Fig. 2. Segmentations of models from the TOSCA high-resolution 3D database using weighted k-means clustering of the proposed feature vectors. Weighted k-means was used to compensate for large variations in local mesh density.

The dataset of 3D surfaces that we wish to segment consists of M surfaces, each composed of N vertices $\mathbf{s}_{ij}$, $i \in \{1, \ldots, N\}$, $j \in \{1, \ldots, M\}$. Denote by $\boldsymbol{\mu}_i = \frac{1}{M} \sum_{j=1}^{M} \mathbf{s}_{ij}$ the mean position of the i-th vertex, averaged over all surfaces. Then $d_{ij} = ||\mathbf{s}_{ij} - \boldsymbol{\mu}_i||_2$ is the euclidean distance from the i-th vertex of the j-th surface to its mean position. The reason why we prefer to work with distance values rather than displacement vectors is because we want vertices to be clustered together even if they move in different directions w.r.t. their mean positions. For example, consider two vertices located on opposite sides of the nose. If the training set contains faces with noses of varying sizes, then these vertices will move further apart or closer together, causing them to have different displacement vectors. Since such a scaling operation could be represented by a single basis vector in the eigenfeatures approach, it is more efficient to assign both vertices to the same region. Based on similar reasoning, we choose not to work with the distances directly, but rather with the normalized distance values

$$t_{ij} = \frac{d_{ij}}{\sqrt{\sum_{l=1}^{M} d_{il}^2}} \tag{1}$$

where the normalization is performed w.r.t. the entire range of displacements the vertex undergoes throughout the training set. The normalized vectors $\mathbf{t}_i = [t_{i1}, \ldots, t_{iM}]^{\mathrm{T}}, i \in \{1, \ldots, N\}$ can now be used as feature vectors for determining similarities between vertices across the training set. By applying a suitable clustering algorithm to the feature vectors, a segmentation into regions of maximum deformation similarity can be obtained. In our experiments, we used the *k-means++* algorithm [25] with 1000 random restarts. The effect of using different types of feature vectors is illustrated in Fig. 1 for an artificially generated dataset. Results on models from the TOSCA high-resolution 3D database [26] are shown in Fig. 2. For the face dataset described in Section 2.1, we obtained the facial components shown in Fig. 3.

Fig. 3. Facial components found by applying the k-means++ clustering algorithm to the similarity features described in Section 3. The features were computed on the 3D shape of a dataset of 187 registered laser scans of the human face. The red box (*right*) shows manual segmentations of the face into four and five regions as used in our experiments (Section 5.1). The manual subdivisions correspond well with those commonly found in the literature.

4 Optimal Region Blending

While a subdivision of an object class into disjoint parts can be useful in itself, it is not enough for optimal part-based reconstructions of a particular object. Consider a part-based 3D shape model of an object. If one of the model parts is allowed to change shape while the rest of the model remains constant, discontinuities are likely to occur at the boundaries between the morphing part and the rest of the model. This is counterproductive in at least two ways:

1. If not properly taken care of, such discontinuities will show up as visible artifacts in the reconstructed object shape. Traditionally [7,11,13,14], this is resolved by blending at the boundaries in a post-processing step.
2. When a part-based model is used in an automatic fitting algorithm, discontinuity errors at the boundaries will be taken into account in the objective function. This may steer the optimization away from the optimal solution, towards a solution that provides a better fit near the boundaries. This issue is not solved by post-processing.

To address both issues, the basis vectors of the part-based model need to be continuous across the entire object. An easy way to achieve this is by training the model on smoothly overlapping training examples, rather than examples that contain all the information of a single region, and are abruptly cut off beyond the region boundaries. The question that remains is how to design the regions of overlap.

4.1 Algorithm Derivation

In this section, we present an algorithm for automatically determining the optimal regions of overlap for a linear part-based model. Recall from Section 3 that we have a training database of M objects, each sampled at N corresponding vertex locations. Ideally, the vertices should be organized such that vertices

with the same index retain the same physical meaning across all objects. E.g., the vertex located at the tip of the nose should have the same index in all face models of the dataset, and similarly for all other points of the face. To keep the derivation as general as possible, we assume that each vertex can be represented by a D-dimensional vector. Furthermore, without loss of generality we assume that each object is vectorized by vertically concatenating its vertices, and mean-normalized by subtracting the corresponding mean from each vertex. I.e., the j-th object in the training set is represented by the DN-dimensional vector

$$\mathbf{s}_j = [\mathbf{s}_{1j}^{\mathrm{T}}, \ldots, \mathbf{s}_{Nj}^{\mathrm{T}}]^{\mathrm{T}} - [\boldsymbol{\mu}_1^{\mathrm{T}}, \ldots, \boldsymbol{\mu}_N^{\mathrm{T}}]^{\mathrm{T}} \tag{2}$$

The entire training set can then be written in matrix form as

$$\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_M] \tag{3}$$

One of the assumptions of the traditional PCA approach is that a particular instance of an object class can be approximated by a linear combination of training examples. Formally, if $\mathbf{y}$ is a (mean-normalized) vector representing a particular instance of the object class, then $\mathbf{y}$ can be approximated as

$$\mathbf{y} \approx \sum_{i=1}^{M} c_i \mathbf{s}_i \tag{4}$$

This principle can be extended to part-based models by introducing a N-dimensional per-vertex weighting vector $\mathbf{w}_j$ for each region $j \in \{1, \ldots, K\}$. After extending the weighting vectors to the full dimensionality of the training vectors by replicating each element D times (which we shall write as $\boldsymbol{\omega}_j$), we obtain

$$\mathbf{y} \approx \sum_{j=1}^{K} \mathrm{diag}(\boldsymbol{\omega}_j)\mathbf{S}\mathbf{c}_j \tag{5}$$

where $\mathbf{c}_j$ is the vector of linear coefficients corresponding to the j-th part of the object. The weighting vectors $\mathbf{w}_j$ contain per-vertex weights specifying the influence that each of the K regions has on the final position (or value) of each vertex. Note that given the weighting vectors, the coefficient vectors that minimize the reconstruction error in the least squares sense are found as

$$[\mathbf{c}_1^{\mathrm{T}}, \ldots, \mathbf{c}_K^{\mathrm{T}}]^{\mathrm{T}} = [\mathrm{diag}(\boldsymbol{\omega}_1)\mathbf{S}, \ldots, \mathrm{diag}(\boldsymbol{\omega}_K)\mathbf{S}]^{+}\mathbf{y} \tag{6}$$

where the superscript $(^{+})$ denotes the Moore-Penrose pseudoinverse. In the interest of brevity, we introduce the notations

$$\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_K] \tag{7a}$$

$$\mathbf{S}_{\mathbf{W}} = [\mathrm{diag}(\boldsymbol{\omega}_1)\mathbf{S}, \ldots, \mathrm{diag}(\boldsymbol{\omega}_K)\mathbf{S}] \tag{7b}$$

for the matrix containing the region weights and the matrix formed by horizontally concatenating the weighted training vectors.

The objective is now to find the weights that allow us to minimize the expected reconstruction error, given the training set. Formally,

$$\mathbf{W}_{\text{opt}} = \arg\min_{\mathbf{W}} \mathrm{E}_{\mathbf{y}}\left[||\mathbf{y} - \mathbf{S}_{\mathbf{W}}(\mathbf{S}_{\mathbf{W}})^{+}\mathbf{y}||_2^2\right] \tag{8}$$

where, in theory, the expectation should be taken w.r.t. the entire population of possible test vectors. Obviously, we don't have access to all possible test vectors. If we did, the optimal basis vectors would be given by a straightforward PCA and the problem would be solved. Here, we only have the training set available and we will have to base the expectation on what's available in there. The solution we propose is to split the training set in two disjoint parts. One part is used for building the region-based subspaces, while the other part serves as a source for generating out-of-training-set examples. To make maximum use of the available data, and to reduce the danger of overfitting, we propose to randomly reassign vectors to both sets in each iteration of the algorithm. The optimal weights can be iteratively estimated with the following alternating least squares algorithm.

Step 1. Given a $DN \times M_X$ matrix of training vectors $\mathbf{X}$, a $DN \times M_Y$ matrix of test examples $\mathbf{Y}$, and a $N \times K$ matrix of region weights $\mathbf{W}$, we need to find the coefficient matrix $\mathbf{C}$ of dimensions $KM_X \times M_Y$ that optimizes

$$\mathbf{C}^* = \arg\min_{\mathbf{C}} \|\mathbf{Y} - \mathbf{X}_{\mathbf{W}}\mathbf{C}\|_{\mathrm{F}}^2 \tag{9}$$

where we have used the notations in Eqs. (7a) and (7b). The subscript F indicates the Frobenius norm. Similar to Eq. (6), the solution is given by

$$\mathbf{C}^* = (\mathbf{X}_{\mathbf{W}})^{+}\mathbf{Y} \tag{10}$$

Step 2. In the next step, we search for the weight matrix $\mathbf{W}^*$ that optimizes

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \|\mathbf{Y} - \mathbf{X}_{\mathbf{W}}\mathbf{C}^*\|_{\mathrm{F}}^2 \tag{11}$$

given $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{C}^*$. First, note that the difference can be rewritten as

$$\mathbf{Y} - \mathbf{X}_{\mathbf{W}}\mathbf{C}^* = [\mathbf{y}_1, \ldots, \mathbf{y}_{M_Y}] - \sum_{i=1}^{K} \mathrm{diag}(\boldsymbol{\omega}_i)[\mathbf{v}_{i1}, \ldots, \mathbf{v}_{iM_Y}] \tag{12}$$

by forming the example vectors $\mathbf{v}_{ij}$, $i \in \{1, \ldots, K\}$, $j \in \{1, \ldots, M_Y\}$ as linear combinations of the training vectors in $\mathbf{X}$, based on the coefficients in $\mathbf{C}^*$. By examining Eq. (12), it becomes clear that the least squares solution to Eq. (11) can be found by solving N independent linear systems (one system of DM_Y equations and K unknowns per vertex). Therefore, at least $M_Y \geq K/D$ test vectors are needed to find a solution.[1]

[1] In some applications, particularly where multimodal data is involved, it might be beneficial to use different weights for each dimension of the vertices. In that case, the requirement becomes $M_Y \geq K$.

The following additional notes complete the algorithm:

1. The weights computed in Step 2 of the iteration may include negative values, which is not physically meaningful. As is standard practice in the NMF framework [27], a valid weight matrix can be found by setting the negative values to zero after each iteration.
2. By additionally constraining the weights to sum to one for each vertex of the model, we ensure that the resulting set of basis vectors retains the ability to perfectly reconstruct the original training vectors.
3. For high dimensional data—such as high-resolution 3D scans—the algorithm converges rather slowly. In our implementation, this is resolved by using a coarse-to-fine approach with four pyramid levels.

4.2 Final Algorithm

To conclude, the final algorithm as used to generate the results in this paper (Fig. 4) can be stated as follows:

Input:
- Mean-normalized training set **S**.
- Number of desired regions K.

Initialization:
- Compute pyramid representation of **S**.
- Compute initial hard segmentation by applying the method described in Section 3.
- Set initial weights **W** according to the hard segmentation.

Iteration:
- For each level of the pyramid:
  - Iterate until convergence or desired number of iterations reached:
    - Randomly split training set into **X** and **Y**.
    - Compute coefficients **C** (Step 1).
    - Compute weights **W** (Step 2).
    - Set negative entries in **W** to zero.
    - Force **W** to sum to one for each vertex.
  - Upsample **W** to the next pyramid level.

Output: W

5 Evaluation

Since the objective of the proposed algorithms is to minimize the reconstruction error for faces that are not present in the training set, the best way to evaluate their performance is by building region-based models using the output **W**, and experimentally testing the reconstruction accuracy on a test set. An important aspect is to compare the results of using different subdivisions.

First, we need to specify how to build a part-based model starting from a set of regions and a training set. The models used in our experiments were constructed according to the following approach:

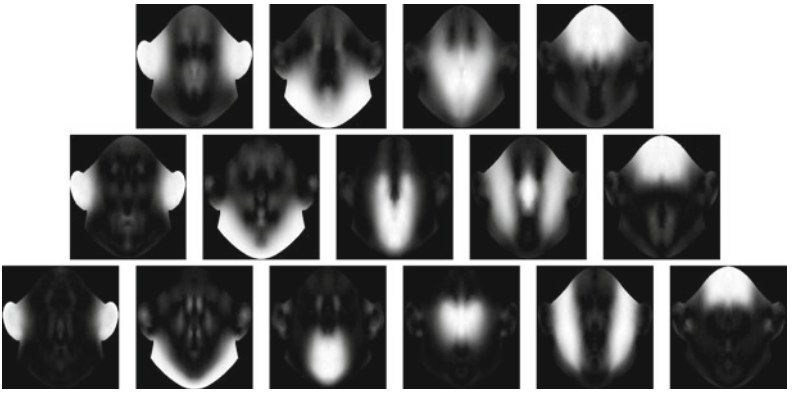


Fig. 4. Optimal weights computed with the algorithm described in Section 4 on a dataset of 187 laser scanned faces, for four (*top*), five (*center*), and six (*bottom*) regions

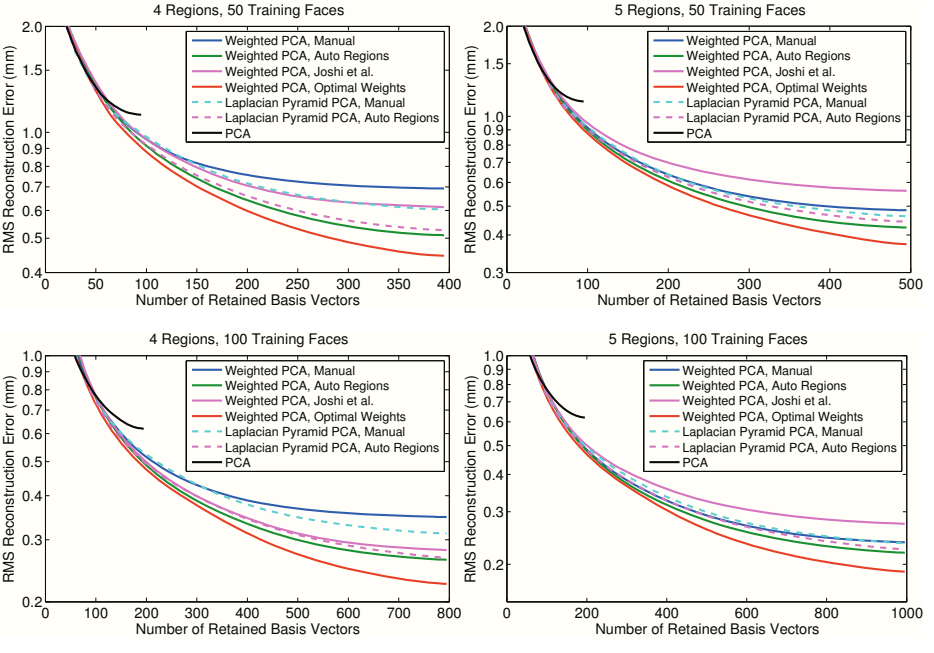


Fig. 5. Average reconstruction errors for the 10-fold cross-validation experiments described in Section 5.1. The standard deviation of the curves is less than 5 μm, and roughly corresponds to the line thickness. Note that the approach using the optimal weights computed with the proposed algorithm (*red solid line*) clearly outperforms the approach using manually segmented regions (*blue solid line*).

1. Create a global training set by subtracting the mean face from the set of training examples.
2. Create region-specific training sets by applying the per-vertex region weights to the global training set (Eq. (7b)).
3. Gather the global and region-specific training sets into a single set and compute basis vectors by applying PCA.

This procedure combines the desirable properties of PCA-based models with the advantages of part-based models. Specifically, the models have the following properties:

- The basis vectors are orthogonal, which avoids redundancy in the model and facilitates the reconstruction task.
- The basis vectors are sorted according to their ability to explain the training data. This ensures that truncating the model by removing some of the least significant basis vectors does not greatly impair the reconstruction quality.
- The model was trained on the set of complete faces as well as their parts, and therefore has knowledge of both global and local deformation statistics. This is in contrast to models trained on only the parts, which would lack any knowledge of statistical relations between the parts.

To avoid discontinuity artifacts at the boundaries for those region-based PCA models that are based on a hard segmentation (either manually or automatically determined), a smooth overlap between regions was created by convolving the hard partition masks with a Gaussian filter having a standard deviation of approximately 10 mm. An alternative approach to the region blending problem that deserves special attention is the method used by Blanz and Vetter [7] for their 3DMM. Instead of simply blending the surface patches at the boundaries according to some weighting factor, they employ a blending technique based on Laplacian pyramids. By simultaneously blending the patches at multiple pyramid levels, a wavelength-dependent overlap size is obtained, which provides discontinuity-free blending, while preserving high-frequency detail. To discriminate between the two blending schemes, we will use the term *Weighted PCA* for all region-based PCA models that use a single matrix $\mathbf{W}$ to define the regions, and *Laplacian Pyramid PCA* for models using different regions of overlap—or equivalently, different weight matrices—for different spatial freqencies.

5.1 Experiments

In our experiments, we compare the reconstruction accuracy on a test set for various part-based models of the 3D shape of human faces. Our experimental setup is as follows. As mentioned in Section 2.1, the dataset consists of 187 laser scanned faces and their mirror images. We test two scenarios: one where 50 faces (100 scans when including the mirror images) are available for training, and one with 100 training faces (200 scans). Ten cross-validation tests are performed. In each test the dataset is randomly partitioned into a training set and a test set (taking care to always assign mirrored scans to the same set as the original ones).

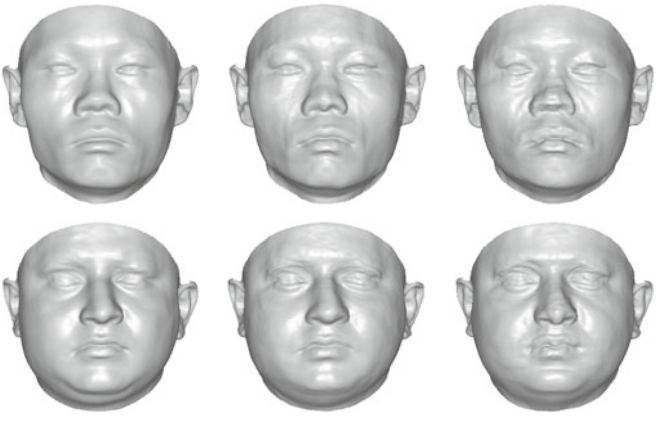


Fig. 6. Examples of face reconstructions based on 50 training faces using four facial regions. *Left:* Original face. *Center:* Reconstructed with a region-based model using optimal region blending (Section 4). *Right:* Reconstruction based on manual regions.

First, a global PCA model is trained, and its reconstruction accuracy is evaluated on the test set for a wide range of model truncations (by model truncation, we mean the operation of retaining the first few basis vectors of a model, while discarding the rest). The process is repeated for region-based models. As a baseline, we manually define segmentations into four components (eyes, nose, mouth, rest) and five components (eyes, nose, mouth, ears, rest) (Fig. 3), and compare models based on these segmentations against the automatically generated ones.

The results are presented in Fig. 5. As expected, the best results were obtained with a weighted PCA approach based on optimal weights computed with the proposed algorithm from Section 4.2. The second-best results were obtained with a weighted PCA model based purely on the automatically clustered regions from Section 3. Given four regions, part-based models based on the method by Joshi et al. [16] still managed to outperform the manual segmentations, while for five regions this method came out last. Our results also show that the performance of Laplacian pyramid PCA models can be improved by using our automatic segmentations. All region-based PCA models in our tests succeeded in outperforming the baseline global PCA from about halfway its number of available basis vectors. Without constraining the number of basis vectors, it is easily feasible to cut the reconstruction error of global PCA in half, or even in three.

Typical examples of the quality improvements that can be expected when upgrading from a manually segmented model to a model with optimal region blending are shown in Fig. 6. Note the overall improved signal-to-noise ratio, and the improved reconstruction quality of facial features (most visible in the nose, mouth, and chin regions).

The main conclusion of these experiments is that part-based models can seriously boost the performance of linear eigenspace-based reconstruction methods, and that there are better alternatives than segmenting the relevant parts by

hand. When compared to other part-based models, the proposed technique provides a significant improvement that is essentially for free, since for a given number of basis vectors the quality improves. Even better: for a given reconstruction accuracy, models using optimally blended regions often get away with less than half the number of basis vectors needed by others.

Given the generality of the derived method, it would be interesting to test it on different modalities, like facial textures, or thermal infrared data. It seems likely that the optimal regions would differ from those obtained for 3D shapes.

6 Conclusion

We have noticed that although the advantages of part-based 3D face representations are widely accepted, the mechanics behind them are only superficially understood. Many state-of-the-art approaches rely on them for enabling lifelike 3D reconstructions, yet none of them seem to have pursued optimality in the design of the regions. Most of the methods are based on manual segmentations, and blending at the boundaries is usually done as a post-processing step. In this paper, we have presented two complementary methods for automatically finding the underlying parts in vectors representing objects of the same class. The first method uses suitable features for finding disjoint regions of maximum deformation similarity, while the second method relaxes the constraints in favor of finding optimal per-vertex weights that minimize the expected reconstruction error on objects outside the training set. In our experiments, the resulting part-based models have been shown to outperform models based on other segmentations.

In future work, it would be interesting to see how these techniques perform on datasets representing other object classes, or the same object class (i.e. faces) seen through different modalities. Also, we intend to check whether the improved reconstruction quality translates to better face recognition results in reconstruction-based approaches.

Acknowledgements. This study was supported by IWT-Flanders through the AMASS++ project (IWT/SBO 060051), and through the project "Forensic Biometric Authentication" (IWT/SBO 60851).

References

1. Biederman, I.: Recognition-by-components: A theory of human image understanding. Psychological Review 94, 115–147 (1987)
2. Brunelli, R., Poggio, T.: Face recognition: Features versus templates. IEEE T. Pattern Anal. 15, 1042–1052 (1993)
3. Pentland, A., Moghaddam, B., Starner, T.: View-based and modular eigenspaces for face recognition. In: Proc. CVPR, pp. 84–91 (1994)
4. Martínez, A.M.: Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class. IEEE T. Pattern Anal. 24, 748–763 (2002)

5. Tarrés, F., Rama, A., Torres, L.: A novel method for face recognition under partial occlusion or facial expression variations. In: Proc. ELMAR, pp. 163–166 (2005)
6. Faltemier, T.C., Bowyer, K.W., Flynn, P.J.: A region ensemble for 3-D face recognition. IEEE T. Inf. Foren. Sec. 3, 62–73 (2008)
7. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3D faces. In: Proc. SIGGRAPH, pp. 187–194 (1999)
8. Peyras, J., Bartoli, A., Mercier, H., Dalle, P.: Segmented AAMs improve person-independent face fitting. In: Proc. BMVC (2007)
9. Blanz, V., Vetter, T.: Face recognition based on fitting a 3D morphable model. IEEE T. Pattern Anal. 25, 1063–1074 (2003)
10. Mian, A., Bennamoun, M., Owens, R.: Region-based matching for robust 3D face recognition. In: Proc. BMVC, vol. 1, pp. 199–208 (2005)
11. Basso, C., Verri, A.: Fitting 3D morphable models using implicit representations. In: Proc. GRAPP, pp. 45–52 (2007)
12. Kakadiaris, I.A., Passalis, G., Toderici, G., Murtuza, M.N., Lu, Y., Karampatziakis, N., Theoharis, T.: Three-dimensional face recognition in the presence of facial expressions: An annotated deformable model approach. IEEE T. Pattern Anal. 29, 640–649 (2007)
13. Zhang, Y., Xu, S.: Data-driven feature-based 3D face synthesis. In: Proc. 3DIM, pp. 39–46 (2007)
14. ter Haar, F.B., Veltkamp, R.C.: 3D face model fitting for recognition. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 652–664. Springer, Heidelberg (2008)
15. Shamir, A.: A survey on mesh segmentation techniques. Comput. Graph. Forum 27, 1539–1556 (2008)
16. Joshi, P., Tien, W.C., Desbrun, M., Pighin, F.H.: Learning controls for blend shape based realistic facial animation. In: Proc. SIGGRAPH (2003)
17. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature 401, 788–791 (1999)
18. Hoyer, P.O.: Non-negative matrix factorization with sparseness constraints. J. Mach. Learn. Res. 5, 1457–1469 (2004)
19. Zass, R., Shashua, A.: Nonnegative sparse PCA. In: NIPS, pp. 1561–1568 (2006)
20. Stauffer, C., Grimson, W.L.: Similarity templates for detection and recognition. In: Proc. CVPR, pp. 221–230 (2001)
21. Basso, C., Paysan, P., Vetter, T.: Registration of expressions data using a 3D morphable model. In: Proc. FG, pp. 205–210 (2006)
22. Kirby, M., Sirovich, L.: Application of the karhunen-loeve procedure for the characterization of human faces. IEEE T. Pattern Anal. 12, 103–108 (1990)
23. Penev, P.S., Sirovich, L.: The global dimensionality of face space. In: Proc. FG, pp. 264–270 (2000)
24. Yang, Q., Ding, X.: Symmetrical PCA in face recognition. In: Proc. ICIP, vol. 2, pp. 97–100 (2002)
25. Arthur, D., Vassilvitskii, S.: k-means++: The advantages of careful seeding. In: Proc. SODA, pp. 1027–1035 (2007)
26. Bronstein, A., Bronstein, M., Kimmel, R.: Numerical Geometry of Non-Rigid Shapes. Springer Publishing Company, Incorporated, Heidelberg (2008)
27. Berry, M.W., Browne, M., Langville, A.N., Pauca, V.P., Plemmons, R.J.: Algorithms and applications for approximate nonnegative matrix factorization. Comput. Stat. Data An. 52, 155–173 (2007)

Color Kernel Regression for Robust Direct Upsampling from Raw Data of General Color Filter Array

Masayuki Tanaka and Masatoshi Okutomi

Tokyo Institute of Technology

Abstract. Upsampling with preserving image details is highly demanded image operation. There are various upsampling algorithms. Many upsampling algorithms focus on the gray image. For color images, those algorithms are usually applied to a luminance component only, or independently applied channel by channel. However, we can not observe the full-color image by a single image sensor equipped in a common digital camera. The data observed by the single image sensor is called raw data. The raw data is converted into the full-color image by demosaicing. Upsampling from the raw data requires sequential processes of demosaicing and upsampling. In this paper, we propose direct upsampling from the raw data based on a kernel regression. Although the kernel regression is known as powerful denoising and interpolation algorithm, the kernel regression has been also proposed for the gray image. We extend to the color kernel regression which can generate the full-color image from any kind of raw data. Second key point of the proposed color kernel regression is a local density parameter optimization, or kernel size optimization, based on the stability of the linear system associated to the kernel regression. We also propose a novel iteration framework for the upsampling. The experimental results demonstrate that the proposed color kernel regression outperforms existing sequential approaches, reconstruction approaches, and existing kernel regression.

1 Introduction

Upsampling with preserving image details is one of highly demanded image operations. The upsampling is sometime called a single image super-resolution, an interpolation, or an inpainting in different context. In this paper we call enlargement of regularly sampled data upsampling. Imaging from irregularly sampled data is refered interpolation. There are various algorithms for the upsampling or the single image super-resolution in the literature [1,2,3,4,5,6]. These upsampling algorithms mainly focus on gray image. For color images, these upsampling algorithms are usually applied to luminance component, while chroma components are upsampled by relatively simple algorithms which include a bilinear interpolation and a joint bilateral upsampling [7]. However, we can not directly observe the luminance component by common consumer digital cameras, since the common consumer digital cameras usually use a single image sensor with a

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 290–301, 2011.

color filter array (CFA). Each pixel element of the single image sensor can record an intensity of one primary color component, typically red, green, or blue. The most popular CFA is the Bayer CFA [8]. The data observed by the single image sensor with the CFA is called raw data. The raw data are then interpolated into a full-color image by a demosaicing process. Then, the luminance component is extracted from the interpolated full-color image. Since the quality of the full-color image strongly depends on the demosaicing process' performance, various demosaicing algorithms have been developed in the literature [9,10,11,12]. In order to obtain upsampled full-color image, we need to apply demosaicing and upsampling sequentially. In this paper, these sequential processes are called sequential approach.

One of other approaches is a reconstruction-based algorithm. A multi-frame color direct super-resolution which reconstructs a high-resolution full-color image from multiple raw data has been proposed [13,14]. Although these reconstruction-based algorithms have been proposed as multi-frame super-resolution, the main idea can be applied to the upsampling or the interpolation. The reconstruction approach with a sparse gradient prior is known as high-performance algorithm [15]. The sparse gradient prior can be also applied to direct upsampling from the raw data.

Recently, kernel regression with an adaptive steering kernel is used in various applications such as denoising and image interpolation [1]. The kernel regression was developed fundamentally for gray images as well as above upsampling algorithms. The kernel regression can interpolate irregularly sampled data. The interpolation of the irregularly sampled data is general algorithm and includes the direct upsampling from the raw data. In this paper, we propose a color kernel regression for robust direct upsampling from the raw data of the general CFA pattern. The proposed color kernel regression can also upsample from the raw data of the general CFA pattern. In the proposed color kernel regression, luminance is modeled by the quadratic polynomial to represent the details, whereas chroma is modeled by the constant polynomial to suppress the color artifacts. We also propose kernel size optimization based on the stability of the linear system associated with the kernel regression. Then, a novel iteration framework for upsampling is presented.

The experimental results demonstrate that the proposed color kernel regression outperforms existing sequential approaches, reconstruction approaches, and existing kernel regression.

2 Interpolation with Kernel Regression

Locally weighted least-squares regression is called kernel regression in [1]. The weighting function is called a kernel and is designed to localize the data. The typical kernel is the Gaussian kernel. Although the goal of this paper is to upsample from the raw data of the general CFA pattern, we discuss the interpolation of irregularly sampled data. The direct upsampling from the raw data is the special case of the interpolation of the irregularly sample data.

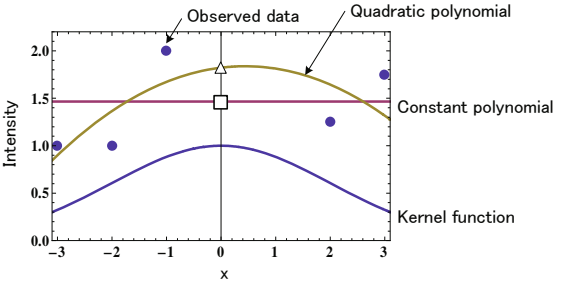


Fig. 1. Interpolation with kernel regression

Let us consider interpolation of the pixel value at the location $\boldsymbol{x}_p$ using irregularly sampled data $\{(\boldsymbol{x}_i, z_i)\}$, where $\boldsymbol{x} = (u, v)^T$ represents a two-dimensional image coordinate, T represents the transpose operator, z is for the sampled pixel value, and suffix i represents the i-th sampling. The pixel value around the location $\boldsymbol{x}_p$ is approximated as

$$\hat{z}(\boldsymbol{x}) = f\left(\boldsymbol{x} - \boldsymbol{x}_p; \hat{\boldsymbol{\theta}}(\boldsymbol{x}_p)\right) , \tag{1}$$

where $f(\boldsymbol{x}; \boldsymbol{\theta})$ is the regression function and $\boldsymbol{\theta}$ is the parameter of the regression function. The parameters $\hat{\boldsymbol{\theta}}(\boldsymbol{x}_p)$ are estimated as

$$\hat{\boldsymbol{\theta}}(\boldsymbol{x}_p) = \arg\min_{\boldsymbol{\theta}} \sum_i k(\boldsymbol{x}_i - \boldsymbol{x}_p) \left[z_i - f(\boldsymbol{x}_i - \boldsymbol{x}_p; \boldsymbol{\theta})\right]^2 ,$$

where $k(\boldsymbol{x})$ is the kernel function. The pixel value at the location $\boldsymbol{x}_p$ is interpolated as

$$\hat{z}\left(\boldsymbol{x}_p\right) = f\left(\boldsymbol{0}; \hat{\boldsymbol{\theta}}(\boldsymbol{x}_p)\right) .$$

In this paper, we consider the Gaussian kernel and the polynomial regression function. We can interpolate the image by performing this pixel interpolation for every necessary pixel location. The case of the constant polynomial (zero-order polynomial) is also known as the Nadaraya–Watson estimator (NWE) [16]. Figure 1 shows the schematic of the interpolation by the kernel regression in one dimensional case. The triangle and the square in Fig. 1 represent the value interpolated with quadratic and constant polynomials, respectively.

2.1 Steering Kernel [1]

Takeda et al proposed an adaptive steering kernel for kernel regression [1]. The steering kernel for the location $\boldsymbol{x}_p$ is represented as

$$k_{\boldsymbol{x}_p}(\boldsymbol{x}) = \exp\left[-\frac{\boldsymbol{x}^T \boldsymbol{C}_{\boldsymbol{x}_p}^{-1} \boldsymbol{x}}{2h^2 \mu_{\boldsymbol{x}_p}^2}\right] ,$$

where $\boldsymbol{C}_{\boldsymbol{x}_p}$ is the covariance matrix of the Gaussian kernel, h is called the global smoothing parameter, and $\mu_{\boldsymbol{x}_p}$ is the local density parameter, which controls the kernel size. The covariance matrix, $\boldsymbol{C}_{\boldsymbol{x}_p}$, is estimated based on the derivatives around the location $\boldsymbol{x}_{\boldsymbol{x}_p}$ as

$$\boldsymbol{C}_{\boldsymbol{x}_p}^{-1} = \frac{1}{|\boldsymbol{N}_{\boldsymbol{x}_p}|} \begin{pmatrix} \sum\limits_{\boldsymbol{x}_j \in \boldsymbol{N}_{\boldsymbol{x}_p}} z_u(\boldsymbol{x}_j) z_u(\boldsymbol{x}_j) & \sum\limits_{\boldsymbol{x}_j \in \boldsymbol{N}_{\boldsymbol{x}_p}} z_u(\boldsymbol{x}_j) z_v(\boldsymbol{x}_j) \\ \sum\limits_{\boldsymbol{x}_j \in \boldsymbol{N}_{\boldsymbol{x}_p}} z_u(\boldsymbol{x}_j) z_v(\boldsymbol{x}_j) & \sum\limits_{\boldsymbol{x}_j \in \boldsymbol{N}_{\boldsymbol{x}_p}} z_v(\boldsymbol{x}_j) z_v(\boldsymbol{x}_j) \end{pmatrix}, \tag{2}$$

where z_u is the horizontal derivative, z_v is the vertical derivative, $\boldsymbol{N}_{\boldsymbol{x}_p}$ is neighbor pixels around the location $\boldsymbol{x}_{\boldsymbol{x}_p}$, and $|\boldsymbol{N}_{\boldsymbol{x}_p}|$ is pixel number of $\boldsymbol{N}_{\boldsymbol{x}_p}$. In [1], the global smoothing parameter is estimated through cross validation, and the local density parameter is estimated as the following [17].

3 Proposed Color Kernel Regression

3.1 Simultaneous Color Kernel Regression

Color images are usually represented by three channels: R, G, and B. For color image processing, the existing kernel regression algorithms [1,18] handle each channel independently. However, it is well known that natural images have strong color correlations, which are used in the color direct super-resolution [13] and in various color demosaicing algorithms [9]. The performance of the kernel regression would be improved if we could involve color correlation into the regression model.

The YCrCb color space is easy to handle color correlation. The luminance, or Y channel, includes high-frequency components, although the chroma, or Cr and Cb channels, include low-frequency components only. Using these properties, we parameterize the regression function in the YCrCb color space as

$$\begin{pmatrix} \mathrm{Y}(u,v) \\ \mathrm{Cr}(u,v) \\ \mathrm{Cb}(u,v) \end{pmatrix} = \boldsymbol{Q}(u,v)\,\boldsymbol{\theta}\,, \tag{3}$$

where $\boldsymbol{\theta}$ is the eight-dimensional vector that represents coefficients of the regression polynomial, and matrix $\boldsymbol{Q}(u,v)$ is expressed as

$$\boldsymbol{Q}(u,v) = \begin{pmatrix} u^2 & uv & v^2 & u & v & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

In this regression model, luminance is modeled using quadratic polynomials to represent the high-frequency components; the chroma are modeled by the constant polynomials to suppress the high-frequency color artifacts.

We presume that the input irregularly sampled color data has only a single channel of pixel value at each sampling point. In order to represent the irregularly sampled color data, we introduce the pixel value vector $\boldsymbol{z}_i$ and the mask vector $\boldsymbol{m}_i$ as

$$\boldsymbol{z}_i = \begin{pmatrix} r_i \\ g_i \\ b_i \end{pmatrix}, \quad \boldsymbol{m}_i = \begin{pmatrix} m_i^r \\ m_i^g \\ m_i^b \end{pmatrix}, \tag{4}$$

where $(r_i, g_i, b_i)^T$ is the pixel value of each channel for i-th sampled data, and $(m_i^r, m_i^g, m_i^b)^T$ is the sampling mask for each channel. The pixel value of the sampled channel represents the sampled value, and the pixel values of non-sampled channels are set to zero. The sampling mask is set to one if the associated color channel is sampled, and is set to zero for the other.

Using the color regression function model presented in Eq. (3) and the sampled data representation in Eq. (4), the proposed color kernel regression is formulated as an optimization problem to minimize the cost function :

$$E_{\boldsymbol{x}_p}(\boldsymbol{\theta}_{\boldsymbol{x}_p}) = \sum_i \left\{ k_{\boldsymbol{x}_p}(\boldsymbol{x}_i - \boldsymbol{x}_p; \mu_{\boldsymbol{x}_p}) \left[\boldsymbol{z}_i - \boldsymbol{R}\boldsymbol{Q}_i^p \boldsymbol{\theta}_{\boldsymbol{x}_p}\right]^T \mathrm{diag}(\boldsymbol{m}_i) \left[\boldsymbol{z}_i - \boldsymbol{R}\boldsymbol{Q}_i^p \boldsymbol{\theta}_{\boldsymbol{x}_p}\right] \right\}, \tag{5}$$

where $\boldsymbol{R}$ is a matrix representing the transformation from the YCrCb color space to the RGB color space, $\mathrm{diag}(\boldsymbol{m})$ is a diagonal matrix whose diagonal elements are elements of $\boldsymbol{m}$, and $\boldsymbol{Q}_i^p$ is defined as

$$\boldsymbol{Q}_i^p = \boldsymbol{Q}(u_i - u_p, v_i - v_p). \tag{6}$$

The covariance matrix of each kernel is calculated by Eq. (2), where the previous estimated luminance component is used for the derivative estimation. The cost function in Eq. (5) is a quadratic form with respect to the regression parameter, $\boldsymbol{\theta}_{\boldsymbol{x}_p}$. Consequently, the optimal solution is obtainable by solving the linear system :

$$\boldsymbol{A}(\mu_{\boldsymbol{x}_p})\,\boldsymbol{\theta}_{\boldsymbol{x}_p} = \boldsymbol{b}(\mu_{\boldsymbol{x}_p}), \tag{7}$$

where

$$\boldsymbol{A}(\mu_{\boldsymbol{x}_p}) = \sum_i k_{\boldsymbol{x}_p}(\boldsymbol{x}_i - \boldsymbol{x}_p; \mu_{\boldsymbol{x}_p}) \left(\boldsymbol{R}\boldsymbol{Q}_i^p\right)^T \mathrm{diag}(\boldsymbol{m}_i) \boldsymbol{R}\boldsymbol{Q}_i^p,$$

$$\boldsymbol{b}(\mu_{\boldsymbol{x}_p}) = \sum_i k_{\boldsymbol{x}_p}(\boldsymbol{x}_i - \boldsymbol{x}_p; \mu_{\boldsymbol{x}_p}) \left(\boldsymbol{R}\boldsymbol{Q}_i^p\right)^T \mathrm{diag}(\boldsymbol{m}_i) \boldsymbol{z}_i.$$

For later discussion, we represent the linear system as a function of the local density parameter $\mu_{\boldsymbol{x}_p}$. Once we obtain the regression parameter $\hat{\boldsymbol{\theta}}_{\boldsymbol{x}_p}$, the interpolated pixel values for each channel of the location $\boldsymbol{x}_p$ are calculated as

$$\begin{pmatrix} \hat{r} \\ \hat{g} \\ \hat{b} \end{pmatrix} = \boldsymbol{R}\boldsymbol{Q}(0,0)\,\boldsymbol{\theta}_{\boldsymbol{x}_p}. \tag{8}$$

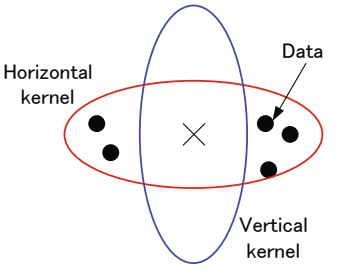


Fig. 2. Kernel shape, kernel size, and data density

3.2 Local Density Parameter Optimization

The local density parameter can control the kernel size. The kernel with the smaller local density parameter is the smaller size of the kernel. Consequently, the kernel with the small local density parameter yields sharp interpolation results. However, the kernel regression with the very small local density parameter is impossible because the coefficient matrix of the linear system in Eq. (7) becomes unstable and singular. When we apply the kernel regression with the small local density parameter for the low data density region, the number of the data which contribute to the regression is insufficient. For this reason, the linear system associated to the kernel with the small local density parameter becomes unstable.

Then, we propose the adaptive local density parameter estimation algorithm. The proposed algorithm estimates the local density parameter, so that the minimum singular value of the coefficient matrix $\boldsymbol{A}(\mu_{\boldsymbol{x}_p})$ equals the stability parameter which is set as small as possible. We manually put 0.01 for the stability parameter. The minimum singular value of the coefficient matrix is known to be one of the stability indexes of the linear system. The proposed algorithm can estimate the minimum local density parameter in the sense of the stability of the linear system in Eq. (7). In this regard, the local density parameter is optimized by the proposed algorithm.

Figure 2 illustrates the schematic relation among the kernel shape, the kernel size, and the data density. The kernel regression with the horizontal kernel is stable because the data exist in a horizontal direction. However, the kernel regression with the vertical kernel is unstable because no data exist in a vertical direction. The stability of the linear system depends on the data density and the kernel shape. The local density parameter estimation based on the data density is insufficient. For these reasons, we measure the stability of the linear system directly to estimate the local density parameter.

3.3 Iterative Color Kernel Regression

In the kernel regression, the derivatives are required to calculate the covariance of the steering kernel as mentioned in Section 2.1. In the proposed color kernel regression, the steering kernel is spatially adapted, but it is common to color

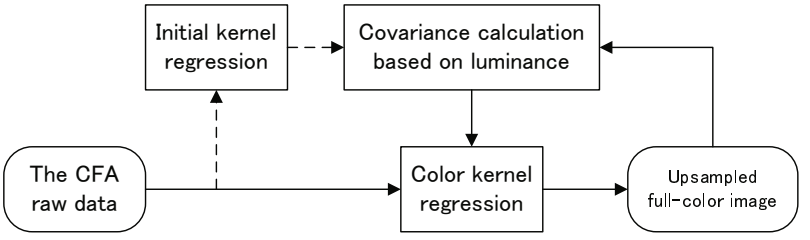


Fig. 3. Block diagram of the proposed iterative color kernel regression, where dashed line represents initial process

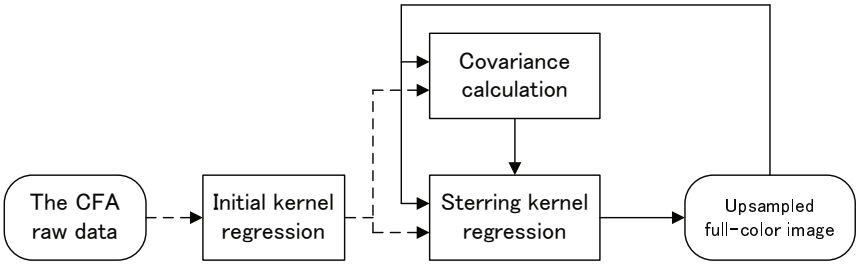


Fig. 4. Block diagram of Takeda's iterative kernel regression [1], where dashed line represents initial process

channels since we simultaneously perform kernel regression for all color channels. For the color kernel regression, the common covariance is calculated based on the derivatives of the luminance component because the luminance component represents the structure or the texture of the image. This is a kind of the chicken and egg problems. In order to handle this problem, we propose an iteration framework. First, we apply initial interpolation for the initial covariance calculation. Then, the covariances of the steering kernels are updated based on the regression result as shown in Fig. 3.

Takeda et al also mentioned an iteration [1]. Figure 4 shows the block diagram of their iteration. The difference between the proposed and Takeda's iteration is the input data of the kernel regression. The input data of the proposed iteration is the observed raw data, while the input data of the Takeda's iteration is the previous upsampled data. In this sense, the upsampling of Takeda's iteration is only performed in the initial kernel regression. The following steering kernel regressions of Takeda's iteration play as denoising. In the contrast, the color kernel regressions of the proposed iteration play direct color upsampling.

4 Experiments

4.1 Direct Upsampling from the Raw Data of the CFA Pattern

Typical six images as shown in Fig. 5 are used as original images for the experimental comparisons. First, the original images are spatially downsampled by

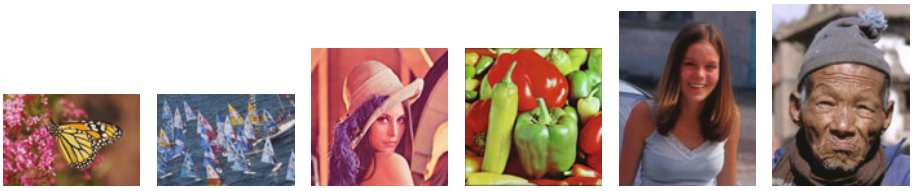
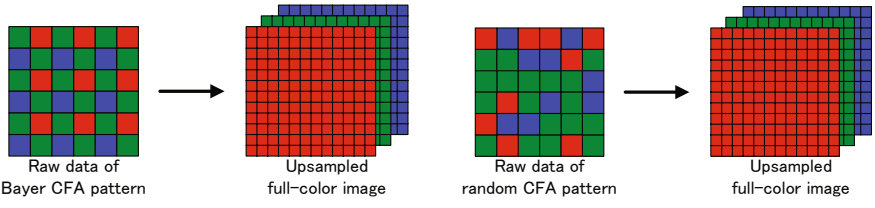


Fig. 5. Six test images



(a) Byaer CFA pattern (b) Random CFA pattern

Fig. 6. Upsampling from raw data of Bayer and random CFA patterns

a factor of two. Then, downsampled images are color-sampled assuming Bayer CFA pattern and random CFA pattern, respectively. A white Gaussian noise is also added to the color-sampled data to synthesize the raw data. We perform upsampling by a factor of two from the synthesized raw data as shown in Fig. 6, so that we restore the same-sized image to the original image.

We compare five algorithms; the proposed color kernel regression, Takeda's kernel regression [1], the gradient sparse prior reconstruction [15], the sequential approach without denoising [6,11], and the sequential approach with denoising [6, 11,19]. The gradient sparse prior reconstruction is performed by minimizing the cost function :

$$I(\boldsymbol{h}) = ||\boldsymbol{z} - \boldsymbol{D}\boldsymbol{h}||_2^2 + \varepsilon \frac{N}{M} ||\Delta \boldsymbol{h}||_{0.8} \,, \tag{9}$$

where $\boldsymbol{z}$ is the vectorized input raw data, $\boldsymbol{h}$ is the vectorized upsampled image to be estimated, $\Delta\boldsymbol{h}$ is the gradient of the upsampled image, $\boldsymbol{D}$ is the matrix which represents downsampling and color-sampling, $\boldsymbol{N}$ is the pixel number of the input raw data, $\boldsymbol{M}$ is the pixel number of the upsampled image, and ε is a regularization parameter. We experimentally set 10^{-8} for the regularization parameter. There are various combinations of demosaicing and upsampling algorithms for the sequential approach. It is infeasible to compare all combinations. In this paper, we use Xiaolin's algorithm [11] for the demosaicing and Xiangjun's algorithm [6] for the upsampling. Since we add the Gaussian noise when the raw data are synthesized, we also apply denoising for the sequential approach. The BM3D algorithm [19] is used for the denoising. The BM3D requires the noise level. We put the true noise level for the BM3D denoising.

Table 1. PSNR comparisons of upsampled images from the noise-free raw data of Bayer CFA pattern

	Monarch	Sail	Lena	Peppers	Girl	Oldman	Average
Proposed color kenel regression	28.58	25.47	29.97	31.45	35.62	31.56	30.44
Takeda's kernel regression [1]	25.65	23.24	27.41	30.25	35.71	31.30	29.02
Gradient sparse prior	24.65	23.03	26.95	29.01	25.42	31.02	28.40
Sequential approach w/o denoising	29.80	26.69	30.32	33.14	37.27	33.07	31.72

Table 2. PSNR comparisons of upsampled images from the noisy raw data of Bayer CFA pattern, where noise level is 10

	Monarch	Sail	Lena	Peppers	Girl	Oldman	Average
Proposed color kenel regression	27.96	24.99	29.02	30.47	33.80	29.08	29.48
Takeda's kernel regression [1]	24.46	22.52	26.10	27.51	29.51	28.00	26.35
Gradient sparse prior	24.01	22.58	26.15	27.44	30.81	28.78	26.63
Sequential approach w/o denoising	26.90	24.92	27.15	28.38	29.36	28.34	27.51
Sequential approach with denoising	26.35	25.22	28.31	29.90	31.69	29.86	28.81

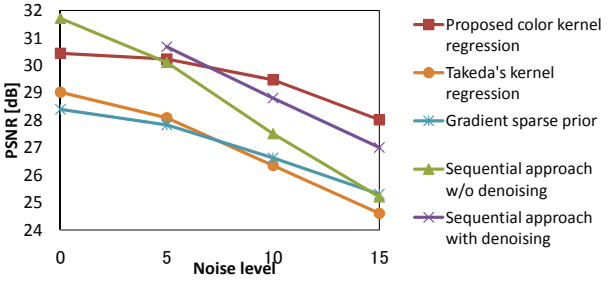


Fig. 7. Average PSNRs of upsampled image from the raw data of Bayer CFA pattern

Table 1 shows the PSNR comparisons of upsampled images from the noise-free raw data of Bayer CFA pattern. The higher PSNR represents that the upsampled image is closer to the original image. In the noise-free case, the sequential approach with denoising is not performed. In this case, the sequential approach shows the highest PSNR for every test images. Table 2 is the PSNR comparisons from the noisy raw data of Bayer CFA pattern, where noise level is 10. In the noisy case, the proposed color kernel regression shows the highest PSNR for many test images. We also evaluate the PSNRs for four noise levels; 0, 5, 15, and 20. The average PSNRs are shown in Fig. 7. These results demonstrate that the proposed color kernel regression can robustly upsample from the raw data of the Bayer CFA pattern compared to the sequential approach with denoising, especially in the high noise case. Note that the demosaicing algorithm [11] of the sequential approach is specialized for the Bayer CFA pattern, while the proposed color kernel regression can be applied to the arbitrary CFA pattern. This is one of advantage of the proposed color kernel regression.

Table 3. PSNR comparisons of upsampled image from the noise-free raw data of random CFA pattern

	Monarch	Sail	Lena	Peppers	Girl	Oldman	Average
Proposed color kenel regression	27.63	24.83	29.53	30.85	36.02	31.94	30.17
Takeda's kernel regression [1]	22.71	21.96	25.52	26.94	33.47	29.52	26.68
Gradient sparse prior	20.64	21.03	24.05	24.36	32.66	28.31	25.17

Table 4. PSNR comparisons of upsampled image from the noisy raw data of radom CFA pattern, where noise level is 10

	Monarch	Sail	Lena	Peppers	Girl	Oldman	Average
Proposed color kenel regression	27.00	24.33	28.61	30.08	33.99	30.81	29.13
Takeda's kernel regression [1]	22.51	21.79	25.14	26.41	31.52	28.63	26.00
Gradient sparse prior	20.54	20.92	23.82	24.10	31.28	27.75	24.74

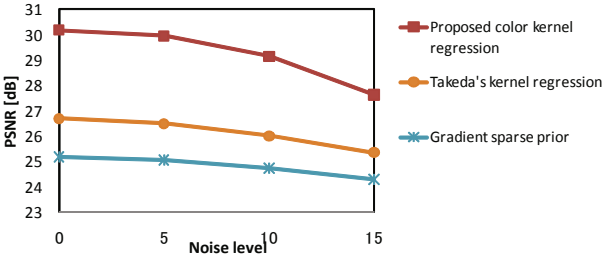


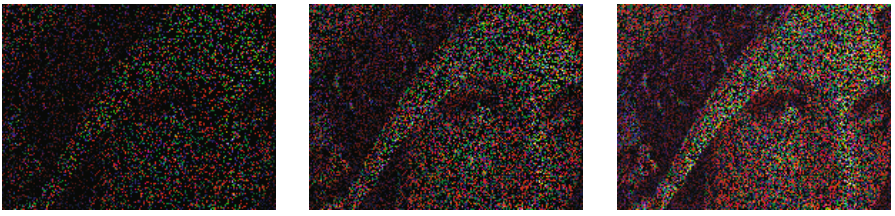
Fig. 8. Average PSNRs of upsampled image from the raw data of random CFA pattern

Next, we consider to upsample the image from the raw data of the random CFA pattern. The random CFA pattern is shown in Fig. 6-(b). We upsample the image with the same manner to those of the Bayer pattern. However, the sequential approach can not be applied since the demosaicing algorithm of the sequential approach is specialized to the Bayer pattern. Tables 3 and 4 show PSNR comparisons of upsampled images from the noise-free and the noisy raw data of the random CFA pattern, respectively. The average PSNRs for each noise level are also shown in Fig. 8. These results demonstrate that the proposed color kernel regression outperforms existing algorithms for the random CFA patterns.

4.2 Interplation from Irregularly Sampled Data

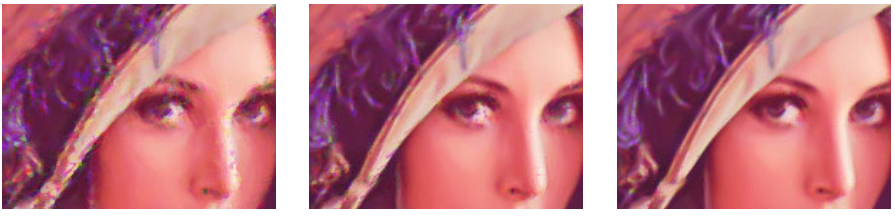
The kernel regression can interpolate irregularly sampled data. We compare the interpolation performance of the proposed color kernel regression and Takeda's kernel regression [1].

Three input data are synthesized from the standard color lena image by irregularly sampling in the space and the color channel, so that their respective data densities are 0.1, 0.2, and 0.3, as presented in Fig. 9. Then, we interpolate color images using the proposed color kernel regression and Takeda's kernel regression.



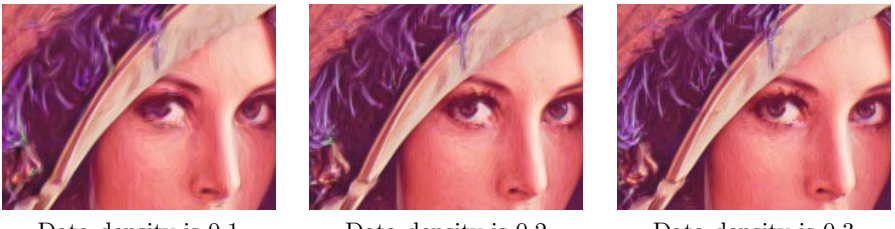
Data density is 0.1 Data density is 0.2 Data density is 0.3

Fig. 9. Input irregularly sampled data



Data density is 0.1 (26.86 [dB]) Data density is 0.2 (29.37 [dB]) Data density is 0.3 (30.54 [dB])

(a) Interpolated images by Takeda's kernel regression [1].



Data density is 0.1 (29.73 [dB]) Data density is 0.2 (32.03 [dB]) Data density is 0.3 (33.45 [dB])

(b) Interpolated image by the proposed color kernel regression.

Fig. 10. Comparisons of the interpolation results. The PSNRs are shown in the subcaption.

Figure 10 shows the interpolation results, where the PSNRs are shown in the subcaption. For all data density case, the proposed color kernel regression shows higher PSNR than Takeda's kernel regression. Interpolation results by Takeda's kernel regression include color artifact, especially in a low data density case. In the contrast, the color artifacts are effectively suppressed by the proposed color kernel regression.

5 Conclusions

We proposed color kernel regression directly to upsample color image from the raw data of the general CFA pattern. The key of the proposed algorithm is to incorporate color correlation into the regression function models. Using the proposed algorithm, the luminance and the chroma are modeled, respectively, as

quadratic and constant polynomials. The other salient point of this proposal is local density parameter optimization based on the stability of the linear system associated to the kernel regression. We have also proposed the iteration framework for the upsampling and the interpolation. The experimental results show that the proposed color kernel regression robustly upsample the color image from the raw data.

References

1. Takeda, H., Farsiu, S., Milanfar, P.: Kernel Regression for Image Processing and Reconstruction. IEEE Transactions on Image Processing 16, 349–366 (2007)
2. Fattal, R.: Image upsampling via imposed edge statistics. ACM Transactions on Graphics (TOG) (26)
3. Shan, Q., Li, Z., Jia, J., Tang, C.: Fast image/video upsampling. In: ACM SIGGRAPH Asia 2008 papers, pp. 1–7. ACM, New York (2008)
4. Freeman, W., Jones, T., Pasztor, E.: Example-based super-resolution. IEEE Computer Graphics and Applications, 56–65 (2002)
5. Glasner, D., Bagon, S., Irani, M.: Super-Resolution from a Single Image. In: IEEE International Conference on Conputer Vision (ICCV) (2009)
6. Zhang, X., Wu, X.: Image Interpolation by Adaptive 2D Autoregressive Modeling and Soft-Decision Estimation. IEEE Transactions on Image Processing 17, 887–896 (2008)
7. Kopf, J., Cohen, M., Lischinski, D., Uyttendaele, M.: Joint bilateral upsampling. ACM Transactions on Graphics (26)
8. Bayer, B.: Color imaging array (1976)
9. Li, X., Gunturk, B., Zhang, L.: Image demosaicing: A systematic survey (Visual Communications and Image Processing) (2008)
10. Gunturk, B., Glotzbach, J., Altunbasak, Y., Schafer, R., Mersereau, R.: Demosaicking: color filter array interpolation. IEEE Signal Processing Magazine 22, 44–54 (2005)
11. Wu, X., Zhang, N.: Primary-consistent soft-decision color demosaicking for digital cameras. IEEE Transactions on image processing 13, 1263–1274 (2004)
12. Hirakawa, K., Parks, T.: Adaptive homogeneity-directed demosaicing algorithm. IEEE Transactions on Image Processing 14, 360–369 (2005)
13. Gotoh, T., Okutomi, M.: Direct super-resolution and registration using raw CFA images. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 600–607 (2004)
14. Farsiu, S., Elad, M., Milanfar, P.: Multiframe demosaicing and super-resolution of color images. IEEE Transactions on Image Processing 15, 141–159 (2006)
15. Levin, A., Fergus, R., Durand, F., Freeman, W.: Image and depth from a conventional camera with a coded aperture. ACM Transactions on Graphics 26, 70 (2007)
16. Nadaraya, E.: On estimating regression. Theory of Probability and its Applications 9, 141 (1964)
17. Silverman, B.: Density estimation for statistics and data analysis (1986)
18. Chatterjee, P., Milanfar, P.: A generalization of non-local means via kernel regression. In: Proc. of SPIE Conf. on Computational Imaging (2008)
19. Dabov, K., Foi, A., Egiazarian, K.: Image restoration by sparse 3D transform-domain collaborative filtering. In: Proc. SPIE Electronic Imaging (2008)

The Large-Scale Crowd Density Estimation Based on Effective Region Feature Extraction Method

Hang Su, Hua Yang, and Shibao Zheng

Institution of Image Communication and Information Processing,
Department of EE, Shanghai Jiaotong University, Shanghai, 200240, China
Shanghai Key Laboratory of Digital Media Processing and Transmission
suhangss@gmail.com, {hyang,sbzh}@sjtu.edu.cn

Abstract. This paper proposes an intelligent video surveillance system to estimate the crowd density by effective region feature extracting (ERFE) and learning. Firstly, motion detection method is utilized to segment the foreground, and the extremal regions of the foreground are then extracted. Furthermore, a new perspective projection method is proposed to modify the 3D to 2D distortion of the extracted regions, and the moving cast shadow is eliminated based on the color invariant of the shadow region. Afterwards, histogram statistic method is applied to extract crowd features from the modified regions. Finally, the crowd features are classified into a range of density levels by using support vector machine. Experiments on real crowd videos show that the proposed density estimation system has great advantage in large-scale crowd analysis. And more importantly, better performance is achieved even on variant view angle or illumination changing conditions. Thus the video surveillance system is more robust and practical.

1 Introduction

Over the past decade, crowd control and management has attracted wide attention from technical and social research disciplines along with the steady population growth and worldwide urbanization. The main reason is on account of a succession of fatal accidents caused by lost control of large-scale crowd around the world, such as the Ivory Coast event during the World Cup Qualifier in 2009 and stampede incident during the Hajj pilgrimage in Saudi Arabia in 2006. To prevent the happening of the sorrowful events, the crowd phenomenon becomes an important research scene in social activities. Among various parameters of crowd phenomenon, crowd density is an important issue of crowd feature analysis, which is closely related to the security level [1]. An intelligent video surveillance system for crowd density estimation is proposed in this paper, which is based on *effective region feature extraction* (ERFE).

The purpose of the crowd density estimation is to analyze the density level of the crowd. Early in 1995, Davies [2] started to estimate the crowd density according to the foreground occupied area ratio in the image. Later, Chow [3]

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 302–313, 2011.

utilized the neural network to analyze the pixel statistical features and obtained the density level of the crowd, which improved the estimation accuracy remarkably. However, these methods work under the assumption that the number of foreground pixels is nearly proportional to the number of people, which is only true when there are no serious occlusions between people. In this case, the effectiveness decreases with increasing density or the obvious occlusion. Alternative methods employed the texture feature of the crowd, such as grey level dependence matrix [4,5], wavelet [6], chebyshev moments [7] to estimate the density of the crowd. Comparing with [3], the methods based on texture could solve the occlusion to some extent, but it is incapable to obtain a desirable result when the density is low. To summarize, previous research failed to obtain an ideal result whin the whole density scope. In addition, if the installation or the visual angle of the camera is changed, the parameters of the estimation model need to be modified, which limits the practicability of the system.

In this paper, the authors present a crowd density estimation system based on region feature extraction. In contrast to the previous methods, the region feature deals with the effective regions rather than all the pixels of the foreground. Logical inference and experimental study both prove that the region features have distinct properties with different density levels of crowd, which makes the system robust in all scale density level. Additionally, the system do not depend on individual detecting or tracking, which is too complicated to implement with heavy crowded scene. Therefore, the system is more suitable and practical for large-scale crowd analysis.

The contributions of this paper are three-fold. Firstly, we present a visual surveillance system which is able to estimate the large-scale crowd density of the input video frames with effective region feature extraction. Secondly, we propose a perspective projection method to modify the 3D to 2D distortion of the extract region, and utilize the regions to fulfill the shadow elimination. Both of the methods improve the accuracy of the system greatly. Finally, we propose a crowd feature extraction method based on region feature. Experiments on real surveillance videos show that the system could achieve a desirable result via analyzing the feature with support vector machine (SVM) [8].

The remainder of this paper is organized as follows: the structure of the system is presented in section 2. Then the details of the crowd estimation algorithm are discussed in section 3, containing the methods to extract and modify the region feature, and the method to extract the crowd feature based on the region feature, then we analyze it with SVM to obtain the density level. In section 4, experiment results of our system will be shown and compared with the traditional methods. Finally, the conclusion is made in section 5.

2 System Architecture

The system mainly includes three modules, which is shown in Figure 1 with different colors: module I denotes the foreground detection; module II and module III denote crowd feature extraction based on region feature and crowd density analysis based on SVM, respectively.

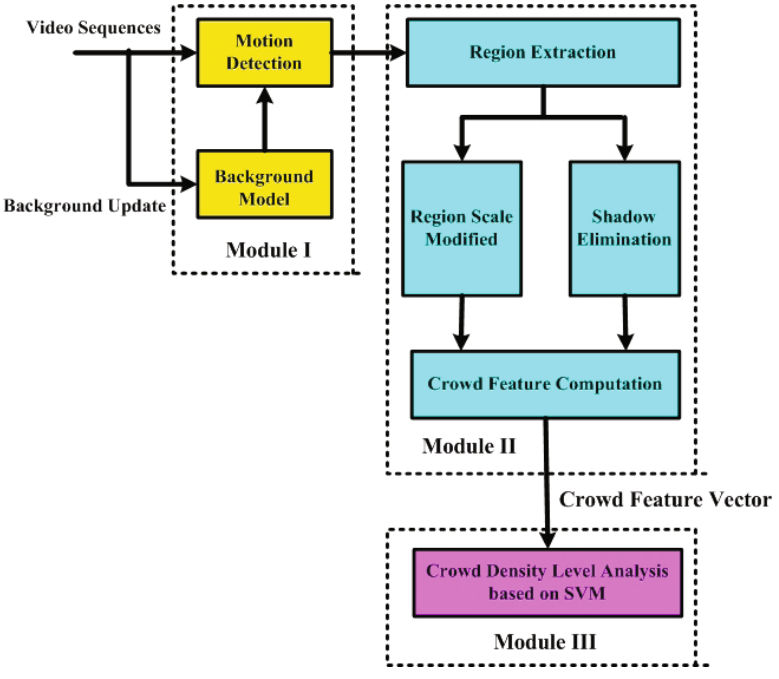


Fig. 1. System Structure of Crowd Density Estimation

Referring to Figure 1, motion detection segments foreground from the background based on mixture-of-Gaussian background modeling and the noise is filtered. Then, the effective region of the foreground are extracted. After that, the corresponding region scale parameters are searched and moving cast shadow is eliminated to improve the accuracy. Then the statistics feature of the regions describing the crowd is exacted. Finally, support vector machine (SVM) is utilized to analyze the crowd density level.

In this paper, we focus our attention on the second module, the crowd feature extraction module. We propose a crowd feature description method based on effective region extraction rather than the individual people segmentation. The main reason are two-fold: first, for highly crowded sites the chances for successfully extracting the effective regions is far above that for segmenting the individuals; second, we are only interested in estimating the crowd density level of the pedestrians and there is no need for separating every people or tracking. Besides, a scale parameter is introduced to modify the region and the influence of moving cast shadow is suppressed with region statistics, both of which improve the performance of the system greatly.

3 Crowd Density Estimation

3.1 Region Feature Extracted and Modified

Region, simply referring to a set of pixels with a certain property, plays an important role in computer vision. The region feature can provide complementary

information about the image, which is not obtained from other descriptors, so we utilize the region feature to represent the crowd. The Maximally Stable Extremal Region (MSER) detector proposed by Matas et al. [9] is the most robust region detector in many cases [10], such as viewpoint or light changing. MSER denotes a set of distinguished regions that are detected in a gray scale image, which is initially used to find correspondences between image elements from two images with different view points. In this paper we employ it as a region extraction prototype for the crowd feature description.

MSER is defined by an intensity function in the region and on its outer boundary, which has two properties: extremal and stable. The extremal property of region $\Re$ implies that

$$\forall p \in \Re, q \in \partial\Re \rightarrow I(p) \geq I(q) \quad or \quad I(p) \leq I(q), \tag{1}$$

where $\partial\Re$ denotes the boundary of region $\Re$, and $I(\cdot)$ denotes the gray level of the pixel. The stable property of region $\Re$ implies that when the threshold ε varies over a large range, there still exists

$$I(p) \geq I(q) \pm \varepsilon \quad or \quad I(p) \leq I(q) \pm \varepsilon, \forall p \in \Re, q \in \partial\Re. \tag{2}$$

MSER can form their superior performance as stable local detector and it can also represent the crowd feature reasonably. In many cases, the gray-level of an unique individual is relatively consistent with a high probability, so the area of the maximally stable extremal regions extracted from an individual should appear to be large and the number of the regions would be relatively small. However, with the density level of the crowd increasing, the occlusion between people would *break* the regions, so the extracted regions will appear to contain less pixels and the number of the regions would be larger. Experiments with different levels of crowd density in Figure 2 further prove the correctness of the logical inference.

In Figure 2, we mask each maximally stable extremal region with different colors in order to distinguish them. From the examples, we can find out that with the increasing of the density level, the number of the extracted regions become larger and the average area of each region become smaller. The problem, which is caused by the typical assumption of previous methods that the number of

Fig. 2. Effective Region Extraction of Different Density Crowd

foreground pixels is approximatively proportional to the density level, is solved by means of extracting crowd feature based on region feature extraction, to great extent. Actually, the region feature could be regarded as the effective pixels in the foreground, and it can also reduce the noise influence that results from background modeling or moving cast shadow, which will be explained later. More accuracy, the traditional MSER need to be modified in two respects, which are focused as follows.

Region Scale Modified. This problem is caused by the 3D to 2D projection, especially for the large-scale crowd. Obviously, object with the same sizes will be smaller in the image when it is farther from the camera, so the region size should be modified in order to have the same measurement before the crowd feature is extracted.

In this paper, we present a method based on perspective projection model. Four restrictive conditions are assumed in the model: all the people are assumed to have similar size; all the people are located in a horizontal plane; the image center and the camera optical center are coincide; and all the pixels in the same row have the same distortion parameter. We will calculate the parameter along each row first.

Under the previous assumptions, all three-dimensional (3-D) lines with a nonzero slope along the optic axis have perspective projections on the image plane that meet at the same point, called the *vanishing point*. Let f be the focal length of the CCD (charge-coupled device) sensor and l_v be the distance between the vanishing point and the center of the CCD sensor. Referring to Figure 3, Θ denotes the horizontal plane and Ω denotes the CCD sensor plane. The visual angle between them is θ, and could be calculated by the equation (3), which is presented in [11]:

$$\theta = \arctan \frac{f}{l_v}. \tag{3}$$

We explain the method in Figure 3 for more details. Suppose there are several equidistant parallel lines which are parallel to the CCD sensor on the horizontal plane in the real world. Referring to Figure 3, $\overrightarrow{AB}$ and $\overrightarrow{CD}$ are the parallel lines in the horizontal plane, which are also parallel to the CCD sensor plane and symmetrical to the optical center. $\overrightarrow{A'B'}$ and $\overrightarrow{C'D'}$ are corresponding lines in the CCD sensor plane. The point M, M', N and N' are intersection points of the vertical plane and the corresponding lines in Θ and Ω. By using similar triangles, the following relation can easily be found:

$$\begin{aligned} \left|\overrightarrow{A'B'}\right| &= \frac{\left|\overrightarrow{FM'}\right|}{\left|\overrightarrow{FM}\right|} \left|\overrightarrow{AB}\right|, \\ \left|\overrightarrow{C'D'}\right| &= \frac{\left|\overrightarrow{FN'}\right|}{\left|\overrightarrow{FN}\right|} \left|\overrightarrow{CD}\right|, \end{aligned} \tag{4}$$

where $\overrightarrow{AA'}$ denotes a line segment begins from point A to point A'. Considering that $\left|\overrightarrow{AB}\right| = \left|\overrightarrow{CD}\right|$, therefore,

$$\frac{\left|\overrightarrow{A'B'}\right|}{\left|\overrightarrow{C'D'}\right|} = \frac{\left|\overrightarrow{FM'}\right|}{\left|\overrightarrow{FM}\right|}\frac{\left|\overrightarrow{FN}\right|}{\left|\overrightarrow{FN'}\right|} \tag{5}$$

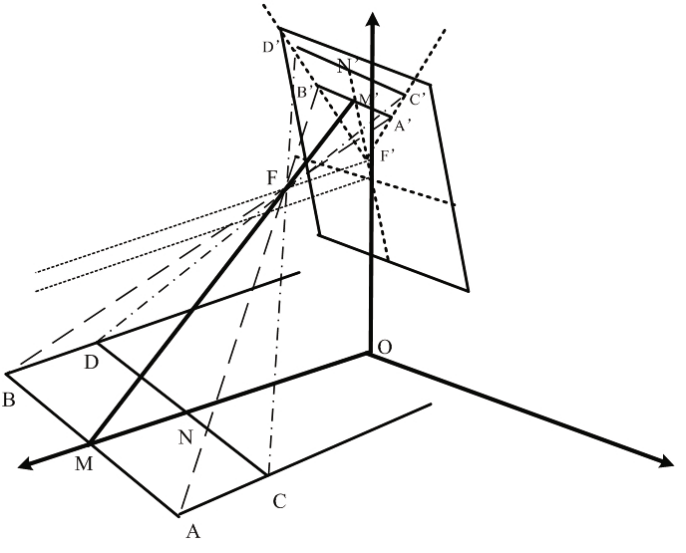


Fig. 3. Model of Horizontal Distortion

Figure 4 shows the vertical plane pass through the optical center. Suppose that the distance between the optic center O of the camera and the ground is h, and the real length of a pixel in the image is μ. According to the geometry analysis, $\frac{\left|\overrightarrow{FM'}\right|}{\left|\overrightarrow{FM}\right|}$ could be calculated as equation (6):

$$\frac{\left|\overrightarrow{FM'}\right|}{\left|\overrightarrow{FM}\right|} = \frac{h_2}{h_1} = \frac{sign(\overrightarrow{OM'})\,|OM'| \cdot \mu \cdot \sin\theta + f \cdot \cos\theta}{h - f \cdot \sin\theta}, \tag{6}$$

where $\overrightarrow{OM}$ points to the objects that are far away from the camera. Obviously, $\frac{\left|\overrightarrow{FN}\right|}{\left|\overrightarrow{FN'}\right|}$could be calculated similarly. Consequently, the distortion parameter along the row could be calculated as equation (7)

$$r_d = \frac{\left|\overrightarrow{A'B'}\right|}{\left|\overrightarrow{C'D'}\right|} = \frac{sign(\overrightarrow{OM'})\left|\overrightarrow{OM'}\right| \cdot \mu \cdot \sin\theta + f \cdot \cos\theta}{sign(\overrightarrow{ON'})\left|\overrightarrow{ON'}\right| \cdot \mu \cdot \sin\theta + f \cdot \cos\theta} \tag{7}$$

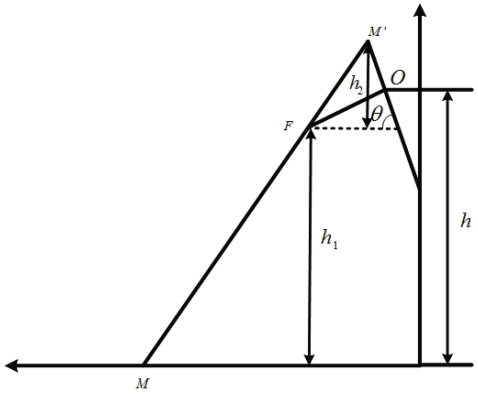


Fig. 4. Chart of Vertical Section through the Focus Point

Given the distortion parameter of the row that pass the center of the image, the parameters in the other rows could be calculated iteratively. In this case, the distortion of the extracted regions $\Re$ could be calculated by surface integral, as equation (8) shows, where N denotes the total pixels number in the regions.

$$w_d = \frac{1}{N} \iint_{\Re} r_d(x, y) d\Re. \tag{8}$$

Shadow Elimination. Shadow is another principal factor that affects the recognition precision of the system. The method proposed in this paper focuses on the moving blobs and analyzes the properties of the blobs in the image. Consequently, if the moving blobs are shadow in fact, the estimation tends to be higher than the practical level. It needs to separate real objects from moving shadows to make the algorithms robust. However, detecting moving shadows is still a challenge in computer vision. Considering the complexity and efficiency, we proposed a shadow elimination method based on the extracted regions.

In order to distinguish between moving cast shadows and moving object points, Jacques et al. [12] proposed a method for detecting shadow after extracting foreground region. A pixel is considered to be shadow pixel if the following condition in equation (9) is satisfied:

$$std_{\Re}(\frac{I_t(x,y)}{\lambda(x,y)}) < L_{std} \quad and \quad L_{low} < (\frac{I_t(x,y)}{\lambda(x,y)}) < 1, \tag{9}$$

where $I_t(x, y)$ denotes the intensity of the pixel (x, y) at frame t, $\lambda(x, y)$ denotes the media of the pixel (x, y). Additionally, $std_{\Re}(\frac{I_t(x,y)}{\lambda(x,y)})$ denotes the standard deviation of quantities $\frac{I_t(x,y)}{\lambda(x,y)}$ over the region $\Re$, and L_{std}, L_{low} are thresholds.

It suggests that the intensity of the pixels in the shadow region tends to be more color invariant than the moving object pixels. Due to the homogeneity of

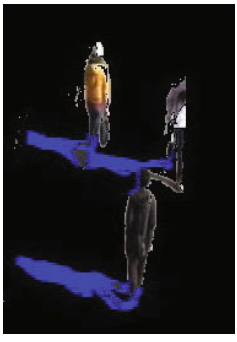


Fig. 5. Shadow Detection based on the Extracted Regions

the intensity of the pixels, the shadows in the image tend to be detected as a larger block of maximally stable extremal region, comparing with the moving people. In this case, the large blocks of shadow that form when the density is low could be detected and eliminated. Although the small blocks of shadow could not be detected effectively when the crowd density is large, they have little influence on the system performance. The example shown in Figure 5 illustrates the result of shadow detection based on the extracted regions, which are marked blue. As it can be observed, the pixels of shadow are correctly identified in the frame.

3.2 Crowd Feature Extraction Based on Region Extraction

As analyzed previously, the size and number of the extracted regions have close relationship with the crowd density. In this section, we compute the modified region size histogram of the extracted regions to obtain the density level of the crowd. Let $H_b(i)$ denotes the count for bin i, $s(i)$ denotes the size for bin i, which is threshold according to different situations. And $M(k)$ denotes the size of every extracted regions, the region size histogram is formed as equation (10):

$$H_b(i) = \{\#M(k) \mid \ s(i) \leq w_d \cdot M(k) \leq s(i+1)\}\,, \tag{10}$$

where $\#M(k)$ denotes the number of the extracted regions that the modified size is between $s(i)$ and $s(i+1)$. Since the sizes of extracted regions vary sharply, ranging from tens of pixels to several thousands, we compute the logarithm of the region size before calculating the histogram to make them concentrate. Therefore, the equation (10) should be modified, that is:

$$H_{\log}(i) = \{\#M_{\log}(k) | s_{log}(i) \leq \log w_d + \log M(k) \leq s_{log}(i+1)\}. \tag{11}$$

Crowd Features of different density levels are shown in Figure 6.

In this paper, we extract the crowd feature of ten dimensions with histogram calculation. The experiment further prove the previous logical inference that the region size and number distinguish with different density level of crowd. Through theoretical analysis and experiment, the small blocks of regions predominate the

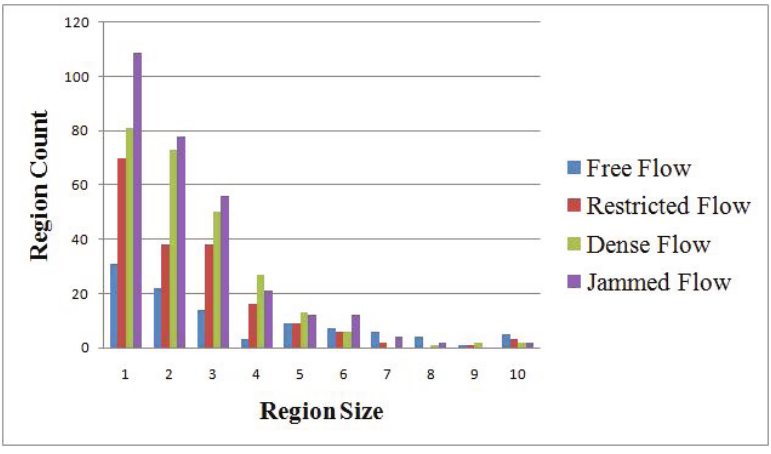


Fig. 6. Crowd Feature of Different Density Level based on Extracted Regions

density level of the crowd, but the regions with a large area above a threshold tend to be shadow. In this case, different dimensions of the crowd feature have different impact on the estimation result.

Beyond that, the extracted regions based on maximally stable extremal region can obtain good result with viewpoint, scale or light changing [10], which makes the system more robust and practical.

3.3 Density Estimation Based on Support Vector Machine

In this section, we will establish the relationship between the crowd feature vector and the output density, which is a typical regression problem. The support vector method [8] is a powerful tool to solve the nonlinear regression estimation problem. The traditional decision function of SVM is shown in equation (12):

$$f(\bar{x}) = sign(\sum_{i=1}^{l} \alpha_i K(\bar{x}_i, \bar{x}) + b), \tag{12}$$

where α_i is Lagrangian multiplier, and $\bar{x}_i$ are support vectors. In this paper we employ the gaussian RBF function as kernel function, as equation (13) shows:

$$K(\bar{x}_i, \bar{x}) = \exp(-\frac{\parallel \bar{x}_i - \bar{x} \parallel^2}{2\sigma^2}). \tag{13}$$

The support vector machines are originally designed for binary classification, but the crowd density estimation is a multi-class problem. Therefore, there is a need to extend it for multi-class problem. Considering the computation complexity and the feature vector property, the one-against-one method [13] is introduced in the crowd density system. This method constructs $k(k-1)/2$ classifiers where each one is trained on data from two classes, and then utilizes the *MaxWins* strategy to decide the density level of the crowd.

4 Experiments and Results

In this work, the video sequences are captured from a camera located with an angle less than 45° towards the ground, which is common position setting in video surveillance applications. The crowd images are first separated into four groups according to the congesting degree of the crowds, i.e., low density ($0 \sim 0.6\text{Peds/m}^2$), moderate low density ($0.6 \sim 1.25\text{Peds/m}^2$), moderate high density ($1.25 \sim 2\text{Peds/m}^2$) and high density ($> 2\text{Peds/m}^2$). These four ranges of crowd densities correspond with the service levels of pedestrian flow, which are defined by Polus [1] as free flow, restricted flow, dense flow and jammed flow. For each crowd density level, we labeled 1000 frames. The system is trained on first 300 frames, and tested on the remaining 700 frames.

In the first experiment, we compare our proposed *crowd feature based on region* with the pixel statistics feature, and the texture feature based on gray level dependence matrix. The estimation result tested on different feature set is shown in Figure 7.

As the Figure 7 shows, *the estimation accuracy based on region feature* performs well on all-scale density level, since it extract the effective blobs of the crowd and the feature between different density crowds differ more obviously. With the density level growing, the accuracy of the pixel statistics feature decreases significantly, because the area and edge ratio could not represent occlusions between people effectively. And for restricted flow or dense flow, the estimation based on texture feature does not have a good result, because the texture feature for these density of crowd is not very distinct.

In addition, we test the proposed system on visual angle varying conditions. For an actual surveillance system, there always exists need to install a pan/tilt to monitor a large scale scene. Besides, the installation location of the camera would also lead to the visual angle varying. All of these conditions would result in the case that the testing samples are captured from a different angle with the training samples. The region extracted based on maximally stable extremal region is invariant with view angle change [10], but the texture feature has a close relationship with the view angle. The result of the estimation result for variant view angle problem is shown in Figure 8.

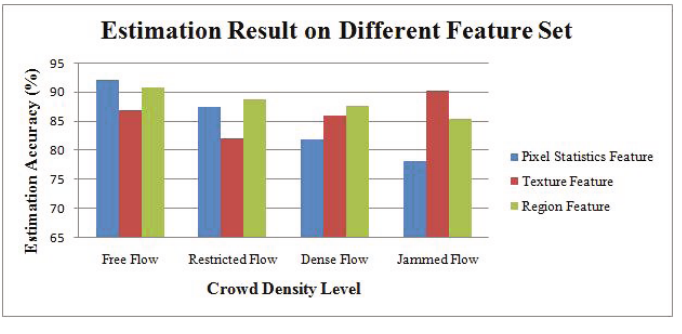


Fig. 7. Estimation Result on Different Feature Set

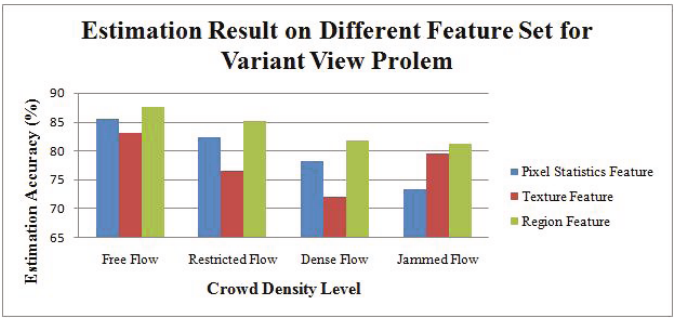


Fig. 8. Estimation Result on Different Feature Set for Variant View Angle Problem

As the figure shows that, the accuracy of crowd density estimation is high for all scale density, compared with the pixel statistics feature and the texture feature.

5 Conclusion

In this paper, the authors propose a system to estimate the crowd density map for the input video frame, which is essential for crowd management in intelligent surveillance system. In this proposed system, the foreground is firstly detected using mixture-of-Gaussian background modeling. Afterwards, the regions are extracted based on maximally stable extremal region. Furthermore, we propose a perspective projection method to modify the 3D to 2D distortion and a moving cast shadow elimination method based on the extracted region. After that, the crowd feature is extracted on the modified region with histogram method. Finally, the system analyze the crowd feature with support vector machine.

Experiments on real videos show that the method proposed in this paper has a good performance within all scale density level crowd. Besides, on account that the region extracted based on maximally stable extremal region is invariant to affine transform and insensitive to light change, the system is more robust and practical than the previous method.

Other than the density, people counting and abnormal detection are also essential issues for crowd surveillance. Therefore, some approaches to estimate the number of people in the scene and understand the crowd behavior are desirable directions in the future.

References

1. Schofer, J., Ushpiz, A., Polus, A.: Pedestrian Flow and Level of Service. J. Transportation Eng. 109, 46–56 (1983)
2. Davies, A.C., Yin, J.H., Velastin, S.A.: Crowd monitoring using image processing. Electronics and Communication Engineering Journal 7, 37–47 (1995)
3. Chow T.W.S., Cho, S.-Y.: Industrial neural vision system for underground railway station platform surveillance. Advanced Engineering Informatics (2002)

4. Wu, X., Liang, G., Lee, K.K., Xu, Y.: Crowd Density Estimation Using Texture Analysis and Learning. In: IEEE International Conference on Robotics and Biomimetics, pp. 214–219 (2006)
5. Sen, G., Wei, L., Ping, Y.H.: Counting people in crowd open scene based on grey level dependence matrix. In: International Conference on Information and Automation, pp. 228–231 (2009)
6. Verona, V.V., Marana, A.N.: Wavelet packet analysis for crowd density estimation. In: Proceedings of the IASTED International Symposia on Applied Informatics, Innsbruck, Austria, pp. 535–540 (2001)
7. Rahmalan, H., Nixon, M.S., Carter, J.N.: On Crowd Density Estimation for Surveillance. In: The Institution of Engineering and Technology Conference on Crime and Security (2006), pp. 540–545
8. Chapelle, O., Haffner, P., Vapnik, V.N.: Support vector machines for histogram-based image classification. IEEE Transactions on Neural Networks 10(5), 1055–1064 (1999)
9. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: British Machine Vision Conference, vol. 1, pp. 384–393 (2002)
10. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Gool, L.: A comparison of affine region detectors. International Journal of Computer Vision 65(1), 43–72 (2005)
11. Pi, M.H., Zhang, H.: Two-Stage Image Segmentation by Adaptive Thresholding and Gradient Watershed. In: Proceedings of the 2nd Canadian Conference on Computer and Robot Vision (CRV 2005), pp. 57–64. IEEE Computer Society, Washington (2005)
12. Jacques, C.S., Jung, C.R., Musse, S.R.: A background subtraction model adapted to illumination changes. In: IEEE Conference on Image Processing, pp: 1817–1820 (October 2006)
13. Kres, R., Ulrich, H.-G.: Pairwise classification and support vector machines. In: Advances in Kernel Methods: Support Vector Learning, pp. 255–268. MIT Press, Cambridge (1999)

TILT: Transform Invariant Low-Rank Textures

Zhengdong Zhang[1], Xiao Liang[1], Arvind Ganesh[2], and Yi Ma[1,2]

[1] Visual Computing Group, Microsoft Research Asia, Beijing
{v-kelviz,v-ollian,mayi}@microsoft.com
[2] Coordinated Science Lab, University of Illinois at Urbana-Champaign
abalasu2@illinois.edu

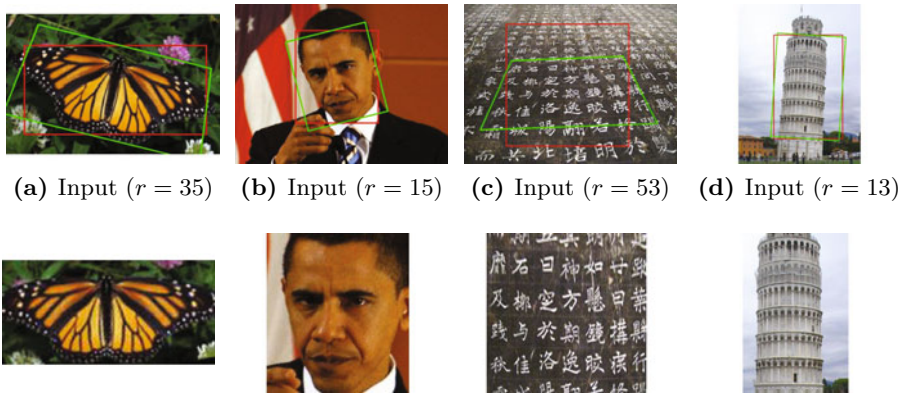
Abstract. In this paper, we show how to efficiently and effectively extract a rich class of low-rank textures in a 3D scene from 2D images despite significant distortion and warping. The low-rank textures capture geometrically meaningful structures in an image, which encompass conventional local features such as edges and corners as well as all kinds of regular, symmetric patterns ubiquitous in urban environments and man-made objects. Our approach to finding these low-rank textures leverages the recent breakthroughs in convex optimization that enable robust recovery of a high-dimensional low-rank matrix despite gross sparse errors. In the case of planar regions with significant projective deformation, our method can accurately recover both the intrinsic low-rank texture and the precise domain transformation. Extensive experimental results demonstrate that this new technique works effectively for many near-regular patterns or objects that are approximately low-rank, such as human faces and text.

1 Introduction

One of the fundamental problems in computer vision is to identify certain feature points or salient regions in images. These points and regions are the basic building blocks of almost all high-level vision tasks such as 3D reconstruction, object recognition, and scene understanding. Throughout the years, a large number of methods have been proposed in the computer vision literature for extracting various types of feature points or salient regions. The detected points or regions typically represent parts of the image which have distinctive geometric or statistical properties such as Canny edges, Harris corners, and textons.

One of the important applications of detecting feature points or regions is to establish correspondence or measure similarity across different images. For this purpose, it is desirable that the detected points/regions are somewhat stable or invariant under transformations incurred by changes in viewpoint or illumination. In the past decade, numerous "invariant" features and descriptors have been proposed, studied, compared, and tuned in the literature (see [1,2] and references therein). A widely used feature descriptor is the *scale invariant feature transform* (SIFT) [3], which to a large extent is invariant to changes in rotation and scale (*i.e.*, similarity transformations) and illumination. Nevertheless, if the images are shot from very different viewpoints, SIFT may fail to establish reliable

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 314–328, 2011.



(a) Input ($r = 35$) **(b)** Input ($r = 15$) **(c)** Input ($r = 53$) **(d)** Input ($r = 13$)

(e) Output ($r = 14$) **(f)** Output ($r = 8$) **(g)** Output ($r = 19$) **(h)** Output ($r = 6$)

Fig. 1. Low-rank Textures Automatically TILTed. From left to right: a butterfly; a face; a tablet of Chinese characters; and the Leaning Tower of Pisa. Top: windows with the red border are the original input, windows with the green border deformed texture returned by our method; Bottom: textures in the green window are matrices of much lower rank.

correspondences and its affine-invariant version becomes a better choice [4,5]. While deformation of a small distant patch can be well-approximated by an affine transform, projective transform becomes necessary to describe the deformation of a large region viewed through a perspective camera. To the best of our knowledge, from a practical standpoint, there are no feature descriptors that are truly invariant (or even approximately so) under projective transformations or homographies.

Despite tremendous effort in the past few decades to search for better and richer classes of invariant features in images, there seems to be a fundamental *dilemma* that none of the existing methods have been able to resolve ultimately: On the one hand, if we consider typical classes of transformations incurred on the image domain by changing camera viewpoint and on the image intensity by changing contrast or illumination, then in strict mathematical sense, *invariants of the 2D image are extremely sparse and scarce* – essentially only the topology of the extrema of the image function remains invariant, known as *attributed Reeb tree* (ART) [6]. The numerous "invariant" image features proposed in the vision literature, including the ones mentioned above, are at best approximately invariant, and often only to a limited extent. On the other hand, *the 3D scene is typically rich of regular structures that are full of invariants* (with respect to 3D Euclidean transformations). For instance, in an urban environment, the scene is typically filled with man-made objects that have parallel edges, right angles, regular shapes, symmetric structures, and repeated patterns. These geometric structures are rich of properties that are invariant under all types of subgroups of the 3D Euclidean group and as a result, their 2D (affine or perspective) images encode extremely rich 3D information about objects in the scene [7,8,9].

In this paper we propose a technique that aims to resolve the above dilemma about invariant features. We contend that instead of trying to seek invariants of the image that are either scarce or imprecise, we should

> *aim to directly detect and extract invariant structures of a scene through their images despite (affine or projective) domain transforms.*

Many methods have been developed in the past to detect and extract all types of regular, symmetric patterns from images under affine or projective transforms (see [10] for a recent evaluation). As symmetry is not a property that depends on a small neighborhood of a pixel, it can only be detected from a relatively large region of the image. However, most existing methods for detecting symmetric regions and patterns start by extracting and putting together local features such as SIFT points [9], corners, and edges [11]. As feature detection and edge extraction themselves are sensitive to local image variations such as noise, occlusion, and illumination change, such symmetry detection methods inherently lack robustness and stability. In addition, as we will see in this paper, many regular structures and symmetric patterns do not even have distinctive features. Thus, we need a more general, effective, and robust way of detecting and extracting regular structures in images despite significant distortion and corruption.

Contributions of this Paper. In this paper, we aim to extract regions in a 2D image that correspond to a very rich class of regular patterns on a planar surface in 3D, whose appearance can be modeled as a "low-rank" matrix. In some sense, many conventional features mentioned above such as edges, corners, symmetric patterns can all be considered as special instances of such low-rank textures. Clearly, an image of such a texture may be deformed by the camera projection and undergoes certain domain transformation (say affine or projective). The transformed texture in general is no longer low-rank in the image. Nevertheless, by utilizing advanced convex optimization tools from matrix rank minimization, we will show how to simultaneously recover such a low-rank texture from its deformed image and the associated deformation.

Our method directly uses raw pixel values of the image and there is no need of any pre-extraction of any low-level, local features such as corners, edges, SIFT, and DoG features. The proposed solution and algorithm are inherently robust to gross errors caused by corruption, occlusion, or cluttered background affecting a small fraction of the image pixels. Furthermore, our method applies to any image regions wherever such low-rank textures occur, regardless of the size of their spatial support. Thus, we are able to rectify not only small local features such as an edge and a corner but also large global symmetric patterns such as an entire facade of a building. We believe that this is a very powerful new tool that allows people to accurately extract rich structural and geometric information about the scene from its images, that are truly invariant of image domain transformations.

Organization of This Paper. The remainder of this paper is organized as follows: Section 2 gives a rigorous definition of "low-rank textures" as well as formulates the mathematical problem associated with extracting such textures. Section 3 gives an efficient and effective algorithm for solving the problem. We provide

extensive experimental results to verify the efficacy of the proposed algorithm as well as the usefulness of the extracted low-rank textures.

2 Transform Invariant Low-Rank Textures

2.1 Low-Rank Textures

In this paper, we consider a 2D texture as a function $I^0(x, y)$, defined on $\mathbb{R}^2$. We say that I^0 is a *low-rank texture* if the family of one-dimensional functions $\{I^0(x, y_0) \mid y_0 \in \mathbb{R}\}$ span a finite low-dimensional linear subspace *i.e.*,

$$r \doteq \dim\left(\text{span}\{I^0(x, y_0) \mid y_0 \in \mathbb{R}\}\right) \leq k \tag{1}$$

for some small positive integer k. If r is finite, then we refer to I^0 as a rank-r texture. Figure 2 shows some ideal low-rank textures: a vertical or horizontal edge (or slope) can be considered as a rank-1 texture; and a corner can be considered as a rank-2 texture. By this definition, it is easy to see that the image of *regular symmetric patterns always lead to low-rank textures.*

Given a low-rank texture, obviously its rank is invariant under any scaling of the function, as well as scaling or translation in the x and y coordinates. That is, if $g(x, y) \doteq cI^0(ax + t_1, by + t_2)$ for some constants $a, b, c, t_1, t_2 \in \mathbb{R}_+$, then $g(x, y)$ and $I^0(x, y)$ have the same rank according to our definition in (1).

For most practical purposes, it suffices to recover any scaled version of the low-rank texture $I^0(x, y)$, as the remaining ambiguity left in the scaling can often be easily resolved in practice by imposing additional constraints on the texture (see Section 3.2). Hence, in this paper, unless otherwise stated, we view two low-rank textures *equivalent* if they are scaled version of each other: $I^0(x, y) \sim cI^0(ax + t_1, by + t_2)$, for all $a, b, c, t_1, t_2 \in \mathbb{R}_+$.

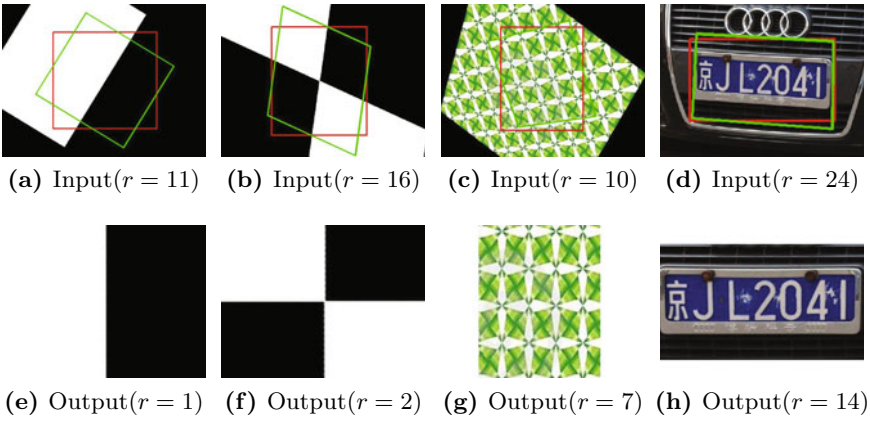
In practice, we are never given the 2D texture as a continuous function in $\mathbb{R}^2$. Typically, we only have its values sampled on a finite discrete, say $m \times n$, grid in $\mathbb{Z}^2$. In this case, the 2D texture $I^0(x, y)$ is represented by an $m \times n$ real matrix. For a low-rank texture, we always assume that the size of the sampling grid is significantly larger than the intrinsic rank of the texture[1] *i.e.*,

$$r \ll \min\{m, n\}$$

Thus, the 2D texture $I^0(x, y)$ (discretized) as a matrix has very low rank relative to its dimensions.

Remark 1 (Low-rank Textures versus Random Textures). Conventionally, the word "texture" is used to describe image regions that exhibit certain spatially stationary stochastic properties (e.g. grass, sand). Such a texture can be considered as a random sample from a stationary stochastic process [12] and is generally of full rank as a 2D function. The "low-rank textures" defined here are complementary to such random textures: It is supposed to describe regions in an image that have rather regular deterministic structures.

[1] It is easy to show that as long as the sampling rate is not one of the aliasing frequencies of the function I^0, the resulting matrix has the same rank as the continuous function.



(a) Input($r = 11$) **(b)** Input($r = 16$) **(c)** Input($r = 10$) **(d)** Input($r = 24$)

(e) Output($r = 1$) **(f)** Output($r = 2$) **(g)** Output($r = 7$) **(h)** Output($r = 14$)

Fig. 2. Representative examples of low-Rank textures. From left to right: an edge; a corner; a symmetric pattern, and a license plate. Top: deformed textures (high-rank as matrices); Bottom: the recovered low-rank textures.

2.2 Deformed and Corrupted Low-Rank Textures

In practice, we typically never see a perfectly low-rank texture in a real image, largely due to two factors: 1. the change of viewpoint usually induces a transformation on the domain of the texture function; 2. the sampled values of the texture function are subject to many types of corruption such as quantization, noise, occlusions, etc. In order to correctly extract the intrinsic low-rank textures from such deformed and corrupted image measurements, we must first carefully model those factors and then seek ways to eliminate them.

Deformed Low-rank Textures. Although many surfaces or structures in 3D exhibit low-rank textures, their images do not! If we assume that such a texture $I^0(x, y)$ lies approximately on a planar surface in the scene, the image $I(x, y)$ that we observe from a certain viewpoint is a transformed version of the original low-rank texture function $I^0(x, y)$:

$$I(x, y) = I^0 \circ \tau^{-1}(x, y) = I^0 \left(\tau^{-1}(x, y)\right)$$

where $\tau : \mathbb{R}^2 \to \mathbb{R}^2$ belongs to a certain Lie group $\mathbb{G}$. In this paper, we assume $\mathbb{G}$ is either the 2D affine group Aff(2) or the homography group $GL(3)$ acting linearly on the image domain[2]. In general, the transformed texture $I(x, y)$ as a matrix is no longer low-rank. For instance, a horizontal edge has rank one, but when rotated by 45°, it becomes a full-rank diagonal edge (see Figure 2(a)).

Corrupted Low-rank Textures. In addition to domain transformations, the observed image of the texture might be corrupted by noise and occlusions or contain some surrounding backgrounds. We can model such deviations as:

$$I = I^0 + E$$

[2] Nevertheless, in principle, our method works for more general classes of domain deformations or camera projection models as long as they can be modeled well by a finite-dimensional parametric family.

for some error matrix E. As a result, the image I is potentially no longer a low-rank texture. In this paper, we assume that only a small fraction of the image pixels are corrupted by large errors, and hence, E is a sparse matrix.

Our goal in this paper is to recover the exact low-rank texture I^0 from an image that contains a deformed and corrupted version of it. More precisely, we aim to solve the following problem:

Problem 1 (Robust Recovery of Transform Invariant Low-rank Textures). Given a deformed and corrupted image of a low-rank texture: $I = (I^0 + E) \circ \tau^{-1}$, recover the low-rank texture I^0 and the domain transformation $\tau \in \mathbb{G}$.

The above formulation naturally leads to the following optimization problem:

$$\min_{I^0, E, \tau} \ \text{rank}(I^0) + \gamma \|E\|_0 \quad \text{subject to} \quad I \circ \tau = I^0 + E \tag{2}$$

where $\|E\|_0$ denotes the number of non-zero entries in E. That is, we aim to find the texture I^0 of the lowest rank and the error E of the fewest nonzero entries that agrees with the observation I up to a domain transformation τ. Here, $\gamma > 0$ is a weighting parameter that trades off the rank of the texture versus the sparsity of the error. For convenience, we refer to the solution I^0 found to this problem as a *Transform Invariant Low-rank Texture* (TILT)[3].

Remark 2 (TILT versus Affine-Invariant Features). TILT is fundamentally different from the affine-invariant features or regions proposed in the literature [4,5]. Essentially, those features are extensions to SIFT features in the sense that their locations are very much detected the same way as SIFT. The difference is that around each feature, an optimal affine transform is found that in some way "normalizes" the local statistics, say by maximizing the isotropy of the brightness pattern [13]. Here TILT finds the best local deformation by minimizing the rank of the brightness pattern in a robust way. It works the same way for any image region of any size and for both affine and projective transforms (or even more general transformation groups that have smooth parameterization). Probably most importantly, as we will see, our method is able to stratify all kinds of regions that are approximately low-rank (e.g. human faces, texts) and the results match extremely well with human perception.

Remark 3 (TILT versus RASL). We note that the optimization problem (2) is strikingly similar to the robust image alignment problem studied in [14], known as RASL. In some sense, TILT is a simpler problem as it only deals with one image and one domain transformation whereas RASL deals with multiple images and multiple transformations, one for each image. Thus, in the next section, we will follow a similar line of development to solve our problem as that in [14]. However, there are some important differences between TILT and RASL. For example, to make TILT work for a wide range of textures, we have to incorporate new constraints so that it achieves a large range of convergence. Moreover, we use a much faster convex optimization algorithm than the APG-based method used in [14], which will be described in the next section.

[3] By a slight abuse of terminology, we also refer to the procedure of solving the optimization problem as TILT.

3 Solution by Iterative Convex Optimization

Although the formulation in (2) is intuitive, the rank function and the ℓ^0-norm are extremely difficult to optimize (in general NP-hard). Recent breakthroughs in convex optimization have shown that under fairly broad conditions, the cost function can be replaced by its convex surrogate [15]: the matrix nuclear norm $\|I^0\|_*$ (sum of all singular values) for $\text{rank}(I^0)$ and the ℓ^1-norm $\|E\|_1$ (the sum of absolute values of all entries) for $\|E\|_0$, respectively. As result, the objective function becomes:

$$\min_{I^0,E,\tau} \|I^0\|_* + \lambda\|E\|_1 \quad \text{subject to} \quad I \circ \tau = I^0 + E \tag{3}$$

where $\lambda > 0$ is a weighting parameter. Notice that although the objective function is now convex, the constraint $I \circ \tau = I^0 + E$ remains nonlinear in $\tau \in \mathbb{G}$. Theoretical considerations in [15] suggest that λ must be of the form $C/\sqrt{\max\{m,n\}}$, where C is a constant, typically set to unity, and $I^0 \in \mathbb{R}^{m\times n}$.

As suggested in [14], to deal with the nonlinear constraint effectively, we may assume that the deformation τ is small and so we can linearize the constraint $I \circ \tau = I^0 + E$ around its current estimate: $I \circ (\tau + \Delta\tau) \approx I \circ \tau + \nabla I \Delta\tau$, where ∇I represents the derivatives of the image w.r.t the transformation parameters[4]. Thus, locally the above optimization problem becomes a convex optimization subject to a set of linear constraints:

$$\min_{I^0,E,\Delta\tau} \|I^0\|_* + \lambda\|E\|_1 \quad \text{subject to} \quad I \circ \tau + \nabla I \Delta\tau = I^0 + E \tag{4}$$

As this linearization is only a local approximation to the original nonlinear problem, we solve it iteratively in order to converge to a (local) minima of the original problem. Although it is difficult to derive exact conditions under which this convex relaxation followed by linearization converges, in practice, we observe that the procedure does converge to a locally optimal solution, even when we start from a large initial deformation τ^0.

3.1 Fast Algorithm Based on Augmented Lagrangian Multiplier

In [14], the accelerated proximal gradient (APG) method was employed to solve the linearized problem (4). Recent studies have shown that the Augmented Lagrangian multiplier (ALM) method [16] is more effective for solving this type of convex optimization problems [15], and typically results in much faster convergence. For the sake of completeness, we will derive the ALM method to the linearized problem (4) and then summarize the overall algorithm for solving the original problem (3). We leave some detailed implementation issues for improving stability and range of convergence to the next subsection.

The Augmented Lagragian Multiplier method aims to solve the original constrained convex program (4) by instead minimizing the augmented Lagrangian given by:

[4] Strictly speaking, ∇I is a 3D tensor: it gives a vector of derivatives at each pixel whose length is the number of parameters in the transformation τ. When we "multiply" ∇I with another matrix or vector, it contracts in the obvious way which should be clear from the context.

$$L(I^0, E, \Delta\tau, Y, \mu) \doteq$$
$$\|I^0\|_* + \lambda\|E\|_1 + \langle Y, I \circ \tau + \nabla I \Delta\tau - I^0 - E\rangle + \frac{\mu}{2}\|I \circ \tau + \nabla I \Delta\tau - I^0 - E\|_F^2$$

where Y is a matrix of Lagrange multipliers, and $\mu > 0$ denotes the penalty for infeasible points. It is known from convex optimization literature [16] that the optimal solution to the original problem (4) can be effectively found by iterating the following two steps till convergence:

$$\begin{cases} (I^0_{k+1}, E_{k+1}, \Delta\tau_{k+1}) & \leftarrow \min_{I^0, E, \Delta\tau} L(I^0, E, \Delta\tau, Y_k, \mu_k) \\ \mu_{k+1} \leftarrow \rho\mu_k, \quad Y_{k+1} & \leftarrow Y_k + \mu_k(I \circ \tau + \nabla I \Delta\tau_{k+1} - I^0_{k+1} - E_{k+1}) \end{cases} \tag{5}$$

for some $\rho > 1$.

In general, it might be expensive to find the optimal solution to the first step of (5) by minimizing over all the variables $I^0, E, \Delta\tau$ simultaneously. So in practice, to speed up the algorithm, we adopt an alternating minimization strategy as follows[5]:

$$\begin{cases} I^0_{k+1} & \leftarrow \min_{I^0} L(I^0, E_k, \Delta\tau_k, Y_k, \mu_k) \\ E_{k+1} & \leftarrow \min_{E} L(I^0_{k+1}, E, \Delta\tau_k, Y_k, \mu_k) \\ \Delta\tau_{k+1} & \leftarrow \min_{\Delta\tau} L(I^0_{k+1}, E_{k+1}, \Delta\tau, Y_k, \mu_k) \end{cases} \tag{6}$$

Given the special structure of our Lagrangian function L, each of the above optimization problem has a very simple solution. Let $\mathcal{S}_t[\cdot]$ be the soft thresholding or *shrinkage* operator defined as follows:

$$\mathcal{S}_t(x) = \text{sign}(x) \cdot \max\{|x| - t, 0\} \tag{7}$$

where $t \geq 0$. When applied to vectors or matrices, the shrinkage operator acts element-wise. Suppose that $(U_k, \Sigma_k, V_k) \doteq \text{svd}(I \circ \tau + \nabla I \Delta\tau_k - E_k + \mu_k^{-1} Y_k)$. Then the optimization problems in (6) can be solved as follows:

$$\begin{cases} I^0_{k+1} & \leftarrow U_k \mathcal{S}_{\mu_k^{-1}}[\Sigma_k] V_k^T \\ E_{k+1} & \leftarrow \mathcal{S}_{\lambda\mu_k^{-1}}[I \circ \tau + \nabla I \Delta\tau_k - I^0_{k+1} + \mu_k^{-1} Y_k] \\ \Delta\tau_{k+1} & \leftarrow (\nabla I^T \nabla I)^{-1} \nabla I^T (-I \circ \tau + I^0_{k+1} + E_{k+1} - \mu_k^{-1} Y_k) \end{cases} \tag{8}$$

We summarize the ALM approach to solving the problem in (3) as Algorithm 1.

3.2 Additional Constraints and Implementation Details

The previous section lays out the basic ALM algorithm for solving the TILT problem (3). However, there are a few caveats in applying it to real images of low-rank textures. In this section, we discuss some additional constraints which make the solution to the problem well-defined and some special implementation details that improve the range of convergence.

[5] It can be shown that under fairly broad conditions, this does not affect the convergence of the algorithm.

Constraints on the Transformations. As we have discussed in Section 2.1, there are certain ambiguities in the definition of low-rank texture. The rank of a low-rank texture function is invariant with respect to scaling in its value, scaling in each of the coordinates, and translation in each of the coordinates. Thus, in order for the problem to have a unique, well-defined optimal solution, we need to eliminate these ambiguities. In the first step of Algorithm 1, the intensity of the image is renormalized at each iteration in order to eliminate the scale ambiguity in pixel value. Otherwise, the algorithm may tend to converge to a "globally optimal" solution by zooming into a black pixel.

To resolve the ambiguities in the domain transformation, we also need some additional constraints. For simplicity, we assume that the support of the initial image window Ω is a rectangle with the length of the two edges being $L(e_1) = a$ and $L(e_2) = b$, so that the total area $S(\Omega) = ab$. To eliminate the ambiguity in translation, we can fix the center x_0 of the window *i.e.*, $\tau(x_0) = x_0$. This imposes a set of linear constraints on $\Delta\tau$ given by :

$$A_t \Delta\tau = 0 \tag{9}$$

To eliminate the ambiguities in scaling the coordinates, we enforce (typically only for affine transforms) that the area and the ratio of edge length remain constant before and after the transformation, i.e. $S(\tau(\Omega)) = S(\Omega)$ and $L(\tau(e_1))/L(\tau(e_2)) = L(e_1)/L(e_2)$. In general, these conditions impose additional nonlinear constraints on the desired transformation τ in problem (3). As outlines earlier, we can linearize these constraints against the transformation τ and obtain another set of linear constraints on $\Delta\tau$:

$$A_s \Delta\tau = 0 \tag{10}$$

As a result, to eliminate both scaling and translation ambiguities, all we need to do is to add two sets of linear constraints to the optimization problem (4).This results in very small modifications to Algorithm 1 to incorporate those additional linear constraints[6].

Multi-Resolution and Branch-and-Bound. To further improve both the speed and the range of convergence, we adopt the popular multi-resolution approach in image processing. For the given image window I, we build a pyramid of images by iteratively blurring and downsampling the window by a factor of 2 until the size of the matrix reaches a threshold (say, less than 30×30 pixels[7]). Starting from the top of the pyramid, we apply our algorithm to the lowest-resolution image first and always initialize the algorithm with the deformation found from the previous level. We found that in practice, this scheme significantly improves the range of convergence and robustness of the algorithm since in the low-resolution images, small details are blurred out and the larger structures of the image drive the updates of the deformation. Moreover, it can speed up Algorithm 1 by

[6] By introducing an additional set of Lagrangian multipliers and then appropriately revising the update equation associated with $\Delta\tau_{k+1}$.

[7] In order for the convex relaxation (3) to be tight enough, the matrix size cannot be too small. In practice, we find that our method works well for windows of size larger than 20×20.

Algorithm 1. (TILT via ALM)

Input: Initial rectangular window $I \in \mathbb{R}^{m\times n}$ in the input image, initial transformations τ in a certain group $\mathbb{G}$ (affine or projective), $\lambda > 0$.
While not converged **Do**
Step 1: normalize the image and compute the Jacobian w.r.t. transformation:

$$I \circ \tau \leftarrow \frac{I \circ \tau}{\|I \circ \tau\|_F}, \quad \nabla I \leftarrow \frac{\partial}{\partial \zeta}\left(\frac{I \circ \zeta}{\|I \circ \zeta\|_F}\right)\Big|_{\zeta=\tau};$$

Step 2: solve the linearized convex optimization (4):

$$\min_{I^0,E,\Delta\tau} \|I^0\|_* + \lambda\|E\|_1 \quad \text{subject to} \quad I \circ \tau + \nabla I \Delta\tau = I^0 + E,$$

with the initial conditions: $Y_0 = 0, E_0 = 0, \Delta\tau_0 = 0, \mu_0 > 0, \rho > 1, k = 0$:
While not converged **Do**

$$\begin{aligned}
(U_k, \Sigma_k, V_k) &\leftarrow \text{svd}(I \circ \tau + \nabla I \Delta\tau_k - E_k + \mu_k^{-1} Y_k),\\
I^0_{k+1} &\leftarrow U_k \mathcal{S}_{\mu_k^{-1}}[\Sigma_k] V_k^T,\\
E_{k+1} &\leftarrow \mathcal{S}_{\lambda\mu_k^{-1}}[I \circ \tau + \nabla I \Delta\tau_k - I^0_{k+1} + \mu_k^{-1} Y_k],\\
\Delta\tau_{k+1} &\leftarrow (\nabla I^T \nabla I)^{-1} \nabla I^T (-I \circ \tau + I^0_{k+1} + E_{k+1} - \mu_k^{-1} Y_k),\\
Y_{k+1} &\leftarrow Y_k + \mu_k (I \circ \tau + \nabla I \Delta\tau_{k+1} - I^0_{k+1} - E_{k+1}),\\
\mu_{k+1} &\leftarrow \rho\mu_k,
\end{aligned}$$

End While
Step 3: update transformations: $\tau \leftarrow \tau + \Delta\tau_{k+1}$;
End While
Output: I^0, E, τ.

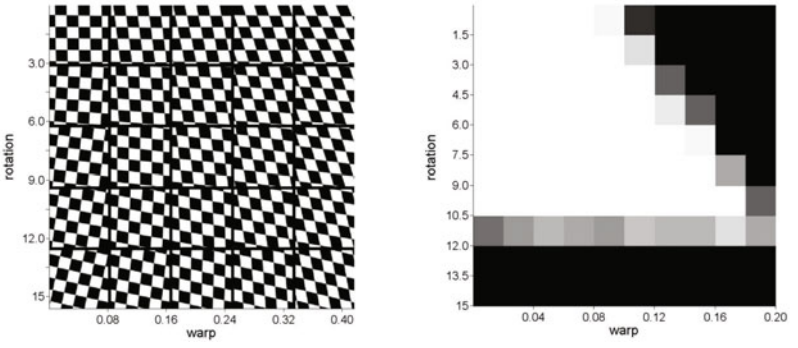


Fig. 3. Convergence of TILT. Left: representative input images in different regions; Right: the range of convergence (# of successes out of 20 random trials in each region)

hundreds of times. We tested the speed of our algorithm in MATLAB on a PC with a 3 Ghz processor. With input matrices of size 50×50, the average running time over 100 experiments is less than 6 seconds.

Apart from the multi-resolution scheme, we can make Algorithm 1 work for a large range of deformation by using a branch-and-bound approach. For instance, in the affine case, we initialize Algorithm 1 with different deformations (e.g., a combination search for all 4 degrees of freedom for affine transforms with no

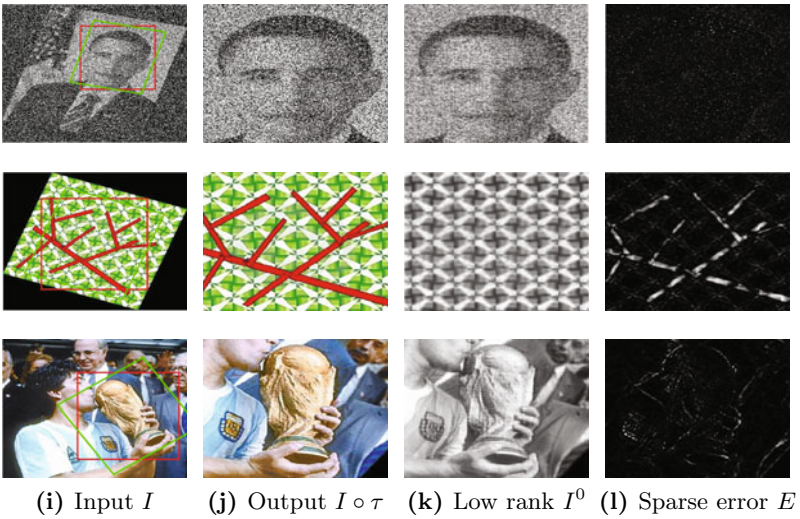
translation). A natural concern about such a branch-and-bound scheme is its effect on speed. Nevertheless, within the multi-resolution scheme, we only have to perform branch-and-bound at the lowest resolution, find the best solution, and use it to initialize the higher resolution levels. Since Algorithm 1 is extremely fast for small matrices at the lowest-resolution level, running multiple times with different initializations does not significantly affect the overall speed. In a similar spirit, to find the optimal projective transform (homography), we always find the optimal affine transform first and then use it to initialize the algorithm. We observed that with such an initialization, the branch-and-bound step becomes unnecesary for the projective transformation case.

Results in all examples and experiments shown in this paper are found by Algorithm 1 using both the multi-resolution and branch-and-bound schemes, unless otherwise stated.

4 Experiments and Applications

4.1 Range of Convergence of TILT

For most low-rank textures, Algorithm 1 has a fairly large range of convergence, even without using any branch-and-bound. To illustrate this, we show the result of the algorithm with a checkerboard image undergoing different ranges of affine transform: $y = Ax + b$, where $x, y \in \mathbb{R}^2$. We parameterize the affine matrix A as $A(\theta, t) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \times \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix}$. We change (θ, t) within the range $\theta \in [0, \pi/6]$ with step size $\pi/60$, and $t \in [0, 0.3]$ with step size 0.03. We observe that the algorithm always converges up to $\theta = 10^\circ$ of rotation and skew (or warp) up to



(i) Input I **(j)** Output $I \circ \tau$ **(k)** Low rank I^0 **(l)** Sparse error E

Fig. 4. Robustness of TILT. Top: random corruption added to 60% pixels; Middle: scratches added on a symmetric pattern; Bottom: containing cluttered background

$t = 0.2$. Due to its rich symmetries and sharp edges, the checkerboard is a challenging case for "global" convergence since there are multiple local minima possible. In practice, we find that for most symmetric patterns in urban scenes (as shown in Figure 5), our algorithm converges for the entire tested range without any branch-and-bound.

4.2 Robustness of TILT

The results shown in Figure 4 demonstrate the striking robustness of the proposed algorithm to random corruptions, occlusions, and cluttered background, respectively. For the first two experiments, the branch-and-bound scheme was not used.

4.3 Shape from Low-Rank Texture Detection

Obviously, the rectified low-rank textures found by our algorithm can better facilitate almost all high-level vision tasks than existing feature or texture detectors, including establishing correspondences among images, recognizing texts and objects, or reconstructing 3D shape or structure of a scene, etc. Due to limited space, we show a few examples in Figure 5 (left) to illustrate how our algorithm can extract rich geometric and structure information from an image of an urban scene.

The image size in this experiment is 1024×685 pixels and we use affine transformations on a grid of 60×60 windows to obtain the low-rank texture. If the rank of the resulting texture drops significantly from that of the original window, we say that the algorithm has "detected" a salient region.[8] In Figure 5, we have plotted the resulting deformed windows, together with the local orientation and surface normal recovered from the optimal affine transformation. Notice that for

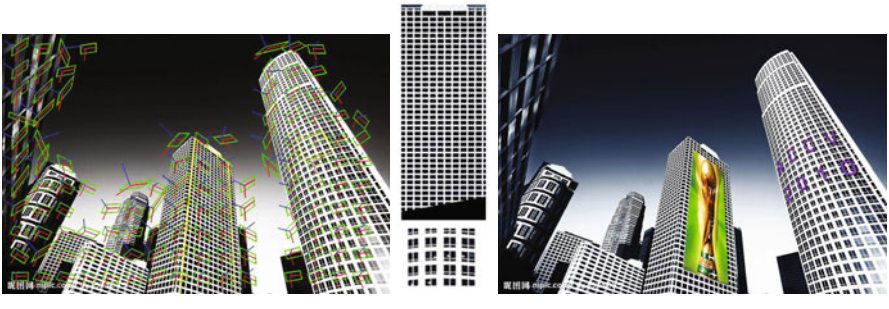


Fig. 5. Left: Low-rank textures detected by the TILT algorithm with affine transform on a grid of 60×60 windows and the recovered local affine geometry. Middle: low rank textures recovered by TILT with projective transform, which correspond to the regions marked with yellow lines; Right: the resulting image with the marked regions edited.

[8] The image rank is computed by thresholding the singular values at 1/30 times the largest one. We also throw away regions whose largest singular value is too small, which typically correspond to a smooth region such as the sky.

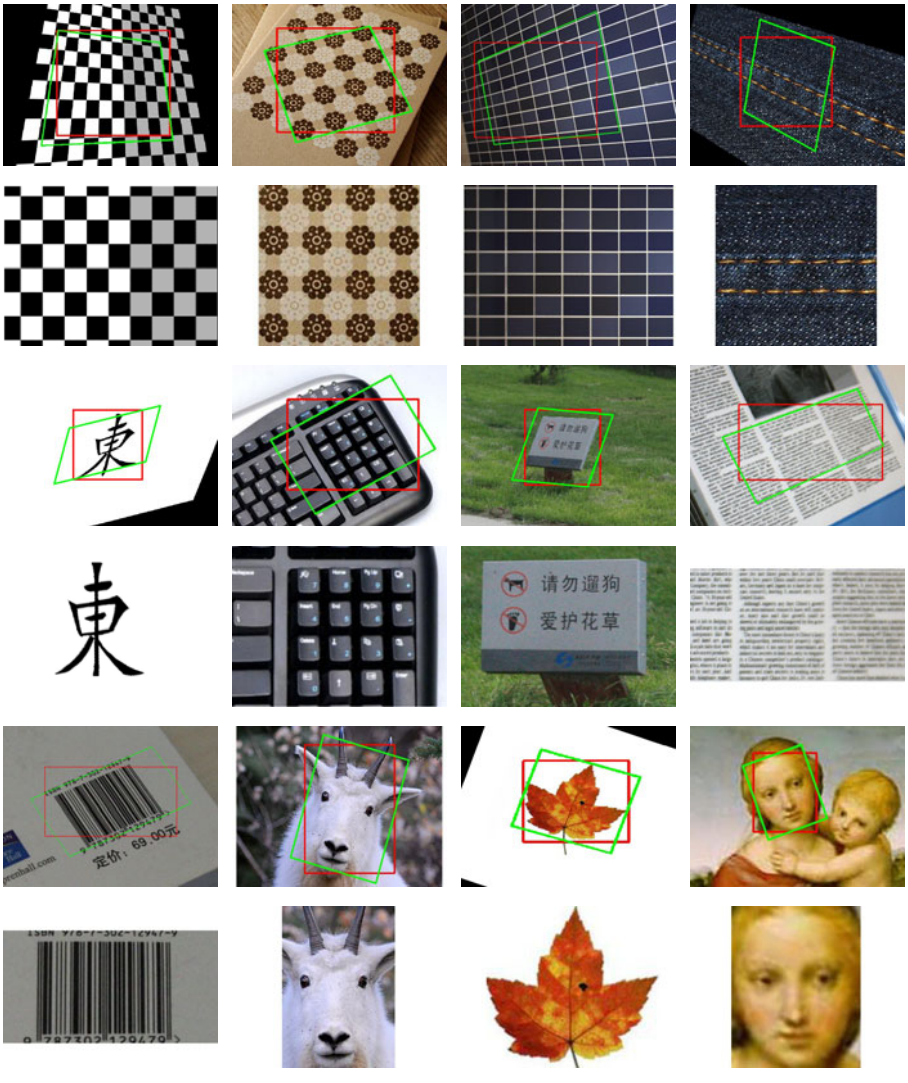


Fig. 6. Representative results of our method. Top: various patterns and textures; Middle: various texts and signs; Bottom: objects with bilateral symmetry.

windows inside the building facades, our algorithm correctly recovers the local geometry for almost all of them; even for patches on the edge of the facades, one of its sides always aligns precisely with the building's edge.

Of course, one can initialize the size of the windows at different sizes or scales. But for larger regions, affine transformations will not be accurate enough to describe the deformation. In this case, we use projective transformations. For

instance, the entire facade of the middle building in Figure 5 (left) obviously exhibits significant projective deformation. Nevertheless, if we initialize the projective TILT algorithm with the affine transform of a small patch on the facade, the algorithm can easily converge to the correct homography and recover the low-rank textures correctly, as shown in Figure 5 (middle).

With both the low-rank texture and their geometry correctly recovered, we can easily perform many interesting tasks such as editing parts of the images using the true 3D orientation and the correct perspective. Figure 5 (right) illustrates this application with an example.

4.4 Rectifying Different Categories of Low-Rank Textures

Since the proposed algorithm has a very large range of convergence for both affine and projective transformations and it is also robust to sparse corruptions, we observed that it works remarkably well for a very broad range of patterns, regular structures, and objects that arise in natural images or paintings. Figure 6 shows a few examples. We observe that with a simple initialization with a very rough rectangular window, our algorithm can converge precisely onto the underlying low-rank structures of the images, despite significant deformation.

Acknowledgements. This work was supported by grants ONR N00014-09-1-0230, NSF CCF 09-64215, and NSF ECCS 07-01676.

References

1. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. PAMI 27, 1615–1630 (2005)
2. Winder, S., Brown, M.: Learning local image descriptor. In: Proc. of CVPR (2007)
3. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV 60, 91–110 (2004)
4. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point descriptors. IJCV 60 (2004)
5. Morel, J.M., Yu, G.: Asift: A new framework for fully affine invariant image comparison. SIAM Journal on Imaging Sciences 2 (2009)
6. Sundaramoorthi, G., Petersen, P., Varadarajan, V.S., Soatto, S.: On the set of images modulo viewpoint and constrast changes. In: Proc. of CVPR (2009)
7. Ma, Y., Soatto, S., Kosecka, J., Sastry, S.: An Invitation to 3D Vision. Springer, Heidelberg (2004)
8. Kosecka, J., Zhang, W.: Extraction, matching, and pose recovery based on dominant rectagular structures. In: CVGIP: Image Understanding, vol. 100, pp. 274–293 (2005)
9. Schindler, G., Krishnamurthy, P., Lublinerman, R., Liu, Y., Dellaert, F.: Detecting and matching repeated patterns for automatic geo-tagging in urban environments. In: Proc. of CVPR (2008)
10. Park, M., Lee, S., Chen, P., Kashyap, S., Butt, A., Liu, Y.: Performance evaluation of state-of-the-art discrete symmetry detection algorithms. In: Proc. of CVPR (2008)

11. Yang, A., Huang, K., Rao, S., Ma, Y.: Symmetry-based 3-D reconstruction from perspective images. In: Computer Vision and Image Understanding, vol. 99, pp. 210–240 (2005)
12. Levina, E., Bickel, P.J.: Texture synthesis and non-parametric resampling of random fields. Annals of Statistics 34, 1751–1773 (2006)
13. Garding, J., Lindeberg, T.: Direct computation of shape cues using scale-adapted spatial derivative operators. IJCV 17, 163–191 (1996)
14. Peng, Y., Ganesh, A., Wright, J., Xu, W., Ma, Y.: RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images. In: Proc. of CVPR (2010)
15. Candès, E., Li, X., Ma, Y., Wright, J.: Robust principal component analysis? (2009) (preprint)
16. Bertsekas, D.P.: Nonlinear Programming. Athena Scientific, Belmont (2004)

Translation-Symmetry-Based Perceptual Grouping with Applications to Urban Scenes

Minwoo Park[1], Kyle Brocklehurst[1], Robert T. Collins[1], 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. An important finding in our understanding of the human vision system is perceptual grouping, the mechanism by which visual elements are organized into coherent groups. Though grouping is generally acknowledged to be a crucial component of the mid-level visual system, in computer vision there is a scarcity of mid-level cues due to computational difficulties in constructing feature detectors for such cues. We propose a novel mid-level visual feature detector where the visual elements are grouped based on the 2D translation subgroup of a wallpaper pattern. Different from previous state-of-the-art lattice detection algorithms for near-regular wallpaper patterns, our proposed method can detect multiple, semantically relevant 2D lattices in a scene simultaneously, achieving an effective translation-symmetry-based segmentation. Our experimental results on urban scenes demonstrate the use of translation-symmetry for building facade super-resolution and orientation estimation from a single view.

1 Introduction

Symmetry is an essential concept in perception and a ubiquitous phenomenon present in all forms and scales in the real world, from galaxies to atomic structures [1]. Symmetry also is considered a preattentive feature [2] that enhances object recognition. Much of our understanding of the world is based on the perception and recognition of repeated patterns that are generalized by the mathematical concept of symmetry [3].

A translation-symmetry is a translation transformation that keeps a pattern setwise invariant [4]. Mathematically, such a pattern has to be periodic and infinite. In practice, we view a finite portion of a periodic pattern in an image as an occluded infinite pattern, thus the term 'translation-symmetry' is equally applicable [5]. 2D translation symmetry detection (lattice detection) has been gaining more attention in computer vision and computer graphics in recent years [5,6,7,8,9,10,11,12,13,14,15,16,17,18]. The underlying topological lattice structure of a near-regular texture (NRT) under a set of geometric and photometric deformation fields was first acknowledged and used by Liu et al. for texture analysis and manipulation [6,19]. Subsequently, Hays et al. [7] developed the first deformed lattice detection algorithm for real images without

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 329–342, 2011.

pre-segmentation. Hays et al. [7] formulated the lattice detection problem as a higher order correspondence problem using a spectral method that produces impressive results. Later, Park et al. [8,9] formulated 2D deformed lattice finding as an inference problem on a Markov Random Field (MRF) and showed improved speed and accuracy on single lattice detection. Regular lattice detection has also been formulated by Han et al. [10] using statistical model selection.

In applications, Shindler et al. [15] use lattice detection to geo-tag user photos and many efforts have been made to remove clutter from real world 2D lattices and synthesize new views [14,20]. Canada et al. [11] developed lattice detection for automatic high throughput analysis of histology array images. Liu et al. [21] apply a lattice detection algorithm to detect and remove a fence region that occludes interesting objects behind the fence.

However, state-of-the-art lattice detection algorithms cannot detect multiple lattices in the scene, which prevents wide applicability of 2D translation symmetry features for many computer vision and graphics applications. In this paper we present, for the first time, an algorithm for detecting multiple 2D lattices.

2 Translation-Symmetry-Based Perceptual Grouping

The human visual system can detect many classes of patterns and statistically significant arrangements of image elements. Perceptual grouping refers to the ability to extract significant image relations and structure from lower-level primitive image features without prior knowledge of high-level image content. Our proposed method follows this concept. We first detect lower-level primitive image features such as Kanade Lucas Tomasi corners (KLT) [22], Maximally Stable Extremal Regions (MSER) [23], and Speeded Up Robust Features (SURF) [24]. Then, each set of feature points is grouped by that feature's descriptor and 2D lattice structures are proposed from each group. The proposed grouping method is an iterative procedure similar to a standard clustering algorithm such as K-means or mean-shift clustering, except that the similarity metric reflects higher-level knowledge of 2D translation symmetry such as texel appearance, (t_1, t_2) basis vector pair, and lattice coverage in the image. Once we obtain this information, we can rectify the perspective distortion of the 2D translation

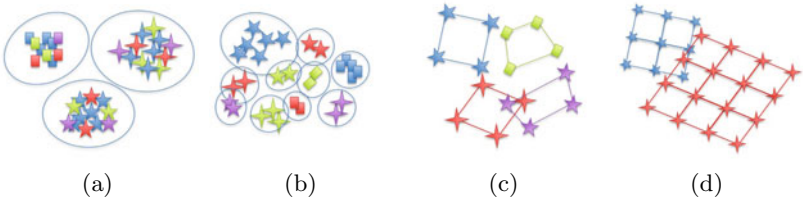
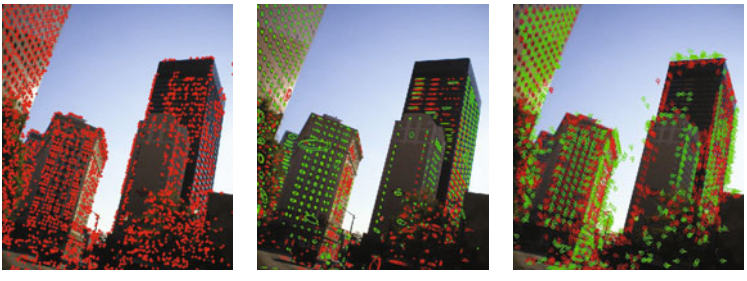


Fig. 1. (a) Lower-level visual primitives (KLT, MSER, and SURF) (b) Visual grouping of each type of feature (c) (t_1, t_2) basis grouping by RANSAC (d) 2D lattice completion and grouping

symmetry as well as collect additional valid lattice points that were not detected by any of the low-level features detectors. This increases the quality of the detected 2D translation symmetry.

2.1 Low-Level Feature Aggregation

The use of different types of lower-level primitive features is beneficial because KLT features, MSER, MSER on the inverted image, and SURF with a positive or negative laplacian generate different responses to different visual elements, and therefore we can reliably find a wide range of 2D lattice points. As can be seen in Figure 2, some of the valid 2D lattice points are only identified by one or two of the detector types, thus justifying using a set of complementary feature detectors.



(a) KLT (b) MSER (c) SURF

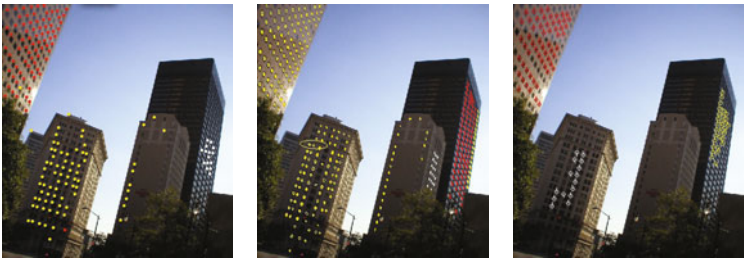
Fig. 2. Low-level primitive visual features detected by KLT, MSER, and SURF. Some of the valid lattice points are not identified by all of the feature detectors but only a subset of them. MSER and MSER on the inverted image are displayed in green and red, respectively, and SURF features with a positive and negative laplacian are colored red and green.

2.2 Grouping of Low-Level Features

Since the number of repeating patterns is not given a priori, we use the mean-shift algorithm with a varying bandwidth to cluster the different types of lower-level features. Since KLT only specifies the 2D location of points and MSER only gives a 2 by 2 scatter matrix of the region, we extract 11 by 11 subimages centered at each KLT feature and the center of the MSER region. Each subimage is normalized by subtracting the mean pixel value, and dividing by the standard deviation of pixel values to compensate for illumination changes.

2.3 Translation-Symmetry-Based Grouping

We seek a (t_1, t_2)-vector pair that represents the generators of the translation symmetry subgroup using a RANSAC-based method, similar to the work of Park et al. [9] and Schindler et al. [15]. Schindler et al. [15] randomly select 4 points from a set of SIFT features, whereas Park et al. [9] improve this random proposal



(a) KLT (b) MSER (c) SURF

Fig. 3. Sample results of mean-shift clustering of low-level features. For clarity we manually choose clusters that are on the 2D lattice structures.

by considering proximity of KLT points to avoid proposals with an invalid affine transformation. We further examine whether the proposed 4 points form a valid quadrilateral to increase the likelihood of finding a feasible perspective mapping. Using this proposal, we iteratively complete the 2D lattice structure under a perspective deformation model while allowing some tolerance using a normalized threshold that is independent of the image.

Proposals of a basis quadrilateral: For each detected feature point cluster, we randomly sample three points $\{a; b; c\}$ to form a (t_1, t_2) vector pair given by $b-a$ and $c-a$, compute the fourth point, d given by t_1+t_2+a, and compute the perspective transformation that maps these four points from image space into the integer lattice basis $\{(0,0),(1,0),(0,1),(1,1)\}$. We can now transform all remaining points from image space into their equivalent lattice positions via the same perspective transform, and count as inlier points those whose lattice space coordinates are within some threshold[1] of an integer position (x, y) . If the four chosen points $\{a; b; c; d\}$ define a valid basis quadrilateral of a 2D translational pattern, many additional supporting votes should emerge from other interest points having a similar spatial configuration.

Lattice completion: Since many of the valid lattice points are not detected by any of the lower-level primitives, we further seek to recover all missed lattice points that are not initially identified by the feature detectors. For this task we evaluate normalized cross correlation between the basis quadrilateral and input image. Note that this is not possible without the hypothesized perceptual grouping of low-level features since otherwise we do not know whether there are repeating patterns, how many there are, and what they look like. Due to possible foreshortening effects, identifying all of the valid lattice points in one iteration using cross correlation suffers from inaccurate localization of the likelihood peaks. To avoid this problem we first rectify the image using the mapping from the current observed lattice points, $\{p_c(j,i)|1 \leq j \leq M, 1 \leq i \leq N\}$ to the

[1] 0.2 is used for all our experiments and this threshold is image independent since all the points are transformed to normalized coordinates (integer coordinates).

(a) Proposals of quadrilateral (b) Final grouping

Fig. 4. Sample results of quadrilateral proposals and final grouping result: (a) Note that there are many duplicate basis quadrilaterals found on the same building. (b) These are further clustered and filtered by the proposed algorithm.

regular lattice constructed by the found (t_1, t_2) basis vector pairs. The (t_1, t_2) basis vector pair and regular lattice point, $p_r(j,i)$ are given by

$$\begin{aligned} t_1 = p_c(1,2) - p(1,1), \quad t_2 = p_c(2,1) - p_c(1,1) \\ p_r(j,i) = p_c(1,1) + t_1(i-1) + t_2(j-1) \end{aligned} \tag{1}$$

We then compute a perspective mapping, H_{cr} from $p_c(j,i)$ to $p_r(j,i)$ and warp input image I_i to get a rectified image $I_r = H(I_i)$. Next, we compute a median quadrilateral, T_m from all the quadrilaterals centered at lattice coordinates $p_r(j,i)$ defined by (t_1, t_2) basis pairs. We compute the normalized cross correlation between median texel T_m and the rectified image I_r $(NCC(I_r, T_m))$ and get local peaks (x, y) by non-maxima suppression.

At this stage, the procedure becomes iterative. We propose a refined mapping, $H_{ri}^{(t)}$ from $p_r^{(t)}(j,i)$ to (j,i) at each iteration t. Only the peaks that are transformed to neighborhoods[2] of integer positions $(\hat{x}, \hat{y})$ are chosen as valid lattice points and used to update the lattice point set $p_r^{(t+1)}(j,i) = p_r^{(t)}(j,i) \cup (\hat{x}, \hat{y})$. We then recompute the rectification mapping $H_{ri}^{(t+1)}$ using correspondences between $p_r^{(t+1)}(j,i)$ and (j,i) and repeat the entire procedure until $p_r^{(t+1)}(j,i) = p_r^{(t)}(j,i)$. This is summarized by pseudo code in Figure 5.

Perceptual grouping of lattices: From all candidate proposals, $\{Pr_i | i = 1 \sim N\}$ we sort all the proposals by the normalized A-score introduced in [6]. The more that quadrilaterals in the lattice look alike and the higher the number of quadrilaterals in the lattice, the smaller the A-score. Starting from the best proposal in terms of the normalized A-score, we group Pr_i while performing the lattice-completion algorithm (section 2.3). As can be seen in Figure 7, the

[2] The same tolerance threshold as section 2.3 is used in all of our experiments.

1: set t=0, Compute (t_1, t_2) and $p_r^{(t)}(j,i)$ by equation (1)
2: Compute mapping H_{cr} using correspondences $p(j,i)$ to $p_r(j,i)$
3: Rectify the input image I_i by H_{cr} to get the rectified image, I_r
4: Compute median quadrilateral T_m
5: Compute normalized cross correlation $NCC(I_r, T_m)$ between T_m and I_r.
6: Compute non-maximum suppressed peaks (x,y) from $NCC(I_r, T_m)$
7: **repeat**
8: compute $H_{ri}^{(t)}$ from $p_r^{(t)}(j,i)$ to (j,i)
9: **if** distance between $H_{ri}^{(t)}$[x;y] and round($H_{ri}^{(t)}$[x;y]) ≤ 0.2 **then**
10: $p_r^{(t+1)}(j,i) = p_r^{(t)}(j,i) \cup (x,y)$
11: **end if**
12: t=t+
13: **until** $p_r^{(t+1)}(j,i) = p_r^{(t)}(j,i)$

Fig. 5. Pseudo code for the lattice-completion algorithm

output of the lattice-completion algorithm (section 2.3) gives rough segmentations of the scene, therefore, we use this information to group the Pr_i. Let the input lattice proposal and output lattice be Pr_i and L_i respectively and let the lattice-completion algorithm (section 2.3) be $F()$, then $L_i = F(Pr_i)$. The initial cluster center, which is a completed 2D lattice, is initialized by $L_1 = F(Pr_1)$ and we then group $\{Pr_i | 2 \leq i \leq N\}$ only when more than 70% of the 2D lattice points in Pr_i are contained in the quadrilaterals in L_1. From the Pr_i that are not grouped to the first cluster center we choose the best proposal in terms of its normalized A-score and we generate a second cluster center using the lattice-completion algorithm (section 2.3). This procedure repeats until no more ungrouped proposals are left. For example, Figure 7a has 72 proposals and the proposed method is successful in grouping all of the proposals. Pseudo code for grouping is given in Figure 6.

2.4 Quantitative Evaluation

We have compared the proposed perceptual grouping algorithm, which we will refer to as **PG**, against Park et al. [9], which we will refer to as **PAMI09**. We have tested the **PAMI09** and **PG** algorithms on a publicly available dataset containing 120 real-world urban scene images with ground-truth [9]. We evaluate the precision and recall rate of the detected lattices using the automated evaluator described in [9]. The number of true positives (TP) is given by the number of correctly identified texels, the number of false positives (FP) is given by the number of falsely detected texels, and the number of false negatives (FN) is given by the number of ground-truth texels minus the number of true positives. When N is the number of 2D lattices in the entire data set, the precision and recall rates are given as

$$Precision = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N}(TP_i + FP_i)}, \quad Recall = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N}(TP_i + FN_i)} \tag{2}$$

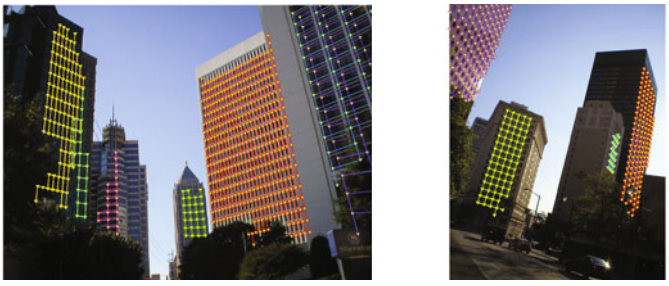
```
For every Pr_i for 1 ≤ i ≤ N
Sort Pr_i by normalized A-score
Enqueue each Pr_i to Q_p, Enqueue(Q_p, Pr_i)
Initialize queue, Q_L for lattice grouping. Q_L = NULL
while Q_p! = NULL do
   L=F(Dequeue(Q_p))
   Enqueue(Q_L,L)
   Initialize temporary queue Q_t = NULL
   while Q_p! = NULL do
      P=Dequeue(Q_p)
      if L ⊃ P then
         Group P to L
      else
         Enqueue(Q_t,P)
      end if
   end while
   Q_p = Q_t
end while
```

Fig. 6. Pseudo code for perceptual grouping

Instead of computing average precision and recall rates, equation (2) is used to reflect the difference between the successful detection of lattices with, for example, 1000 texels versus 4 texels.

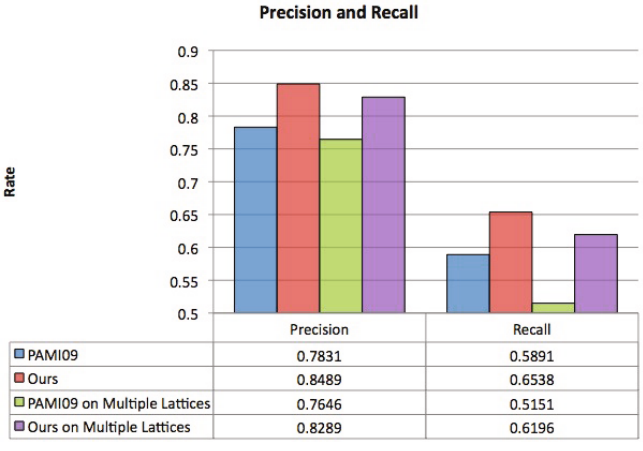
Accuracy: We measure the detection rate of **PAMI09** and **PG** only when these two algorithms detect the same lattice structure, since **PAMI09** [9] is intended for detecting only a single deformed lattice (14672 ground-truth texels). Second, we measure detection rates against all of the ground-truth to show the multiple lattice detection capability of **PG** (23753 number of ground-truth texels). Since **PAMI09** [9] is not intended for multiple lattice detection, we first run **PAMI09** [9], then remove the portion of image where the 2D lattice is found, and repeat until no more lattices are found.

As can be seen in Figure 8, the precision rate of **PG** has improved by 8.4 % over **PAMI09** [9] for both single and multiple lattice detection and the recall rate of **PG** is improved by 10% and 20% over **PAMI09** [9] for single and multiple lattice detection respectively. In addition, the precision and recall rates of **PG** and **PAMI09** [9] for detecting multiple lattices does not drop significantly from the rates on single lattices, as can be seen in Figure 8. This effectiveness of our method comes from: 1) feature aggregation from a variety of interest point detectors, which is more reliable at exposing repeating structures; 2) modeling the deformation of the lattice by perspective projection rather than non-rigid deformation for fast, simple, and accurate application to rigid objects; and 3) perceptual grouping of multiple lattices.



(a) (b)

Fig. 7. Sample results of translation-symmetry-based perceptual grouping are shown. Different colors mean different groups.



	Precision	Recall
PAMI09	0.7831	0.5891
Ours	0.8489	0.6538
PAMI09 on Multiple Lattices	0.7646	0.5151
Ours on Multiple Lattices	0.8289	0.6196

Fig. 8. The blue and red bars indicate precision and recall rate of single lattice detection (14672 ground-truth texels) for the **PG** (red) and the **PAMI09** [9] (blue) algorithms. The green bar indicates precision and recall of sequential runs of **PAMI09** [9] and the purple bar indicates precision and recall rate of **PG** for detecting multiple lattices within a single image (23753 ground-truth texels).

Efficiency: The **PG** algorithm takes 4.2 ± 2.07 min using a 2.4 GHz Intel P8600 4GB machine in MATLAB while **PAMI09** [9] takes 15.8 ± 11.3 min. This confirms that the new method is more efficient and more accurate.

3 Application

To demonstrate a possible application using the 2D lattice grouping proposed in this paper, we have used the detected lattices for single view super-resolution and urban scene analysis.

3.1 Super-Resolution from a Single View

Recently Glasner et al. [25] showed the power of super-resolution from a single view. Recurrence of similar patches in an image forms the basis for their single view super-resolution approach in [25], and therefore correctly identifying corresponding patches is very important in this. As can be seen in our results in Figure 7, we solve this correspondence problem for texels in a lattice structure.

Instead of running a state-of-the-art super-resolution algorithm such as [25], we took a basic approach where multiple images of the same scene are registered, a median image is computed, and de-blurring is performed. In our case we rectify each quadrilateral in the 2D lattice into the same coordinate system, compute a median texel, and perform deconvolution to get a high resolution (HR) image. We map each recovered HR image back to the original space and combine the existing original low resolution (LR) image to transfer high frequency HR information while retaining original lighting and shadow changes. We first perform a discrete cosine transform (DCT) on the HR image to isolate the high frequency components by truncating absolute DCT coefficients larger than 80% of the largest absolute DCT components to get truncated DCT block D[3]. Inverse DCT is then performed to get HR_{hf} and HR_{hf} is added back to the original LR image, thus preserving local information[4]. Sample results are shown in Figure 9.

3.2 Frontal View Facades Estimation from a Single View

Before we attempt to analyze an urban scene, we need to resolve ambiguity of (t_1, t_2) vector pairs under perspective distortion since there could be many choices of valid (t_1, t_2) vector pairs for a given perspective distortion of a 2D wallpaper pattern. This can make estimation of frontal facets of buildings ambiguous. We want the (t_1, t_2) vector pair to be aligned to vertical and horizontal edges of the building, since these edges are typically aligned with meaningful directions, either parallel to or perpendicular with the ground. Figure 11 (a) shows (t_1, t_2) vector pairs that are not aligned with the horizontal and vertical edges of the building. Figure 11 (b) shows (t'_1, t'_2) vector pairs after the desired correction. In the following section we will explain in detail how we correct (t_1, t_2) vector pairs.

Resolving ambiguity of (t_1, t_2) vector pairs: Most modern architecture falls into either the pmm or p4m subgroup of the 17 possible 2D wallpaper patterns [15]. In such cases, the (t_1, t_2) vector pair should be aligned with both the reflection axes and the horizontal and vertical edges of the building[5]. First, we enumerate variations of (t_1, t_2) from the current detected lattice. Let a lattice point at row j and column i be given by $p(j, i)$, then the current t_1 and t_2 are

[3] This is the inverse of JPEG procedure where one wants to discard high frequency information to achieve compression.

[4] For further details, please refer to our supplemental material.

[5] We do not examine horizontal and vertical gradient information to correct (t_1, t_2) as there might be severe perspective distortion.

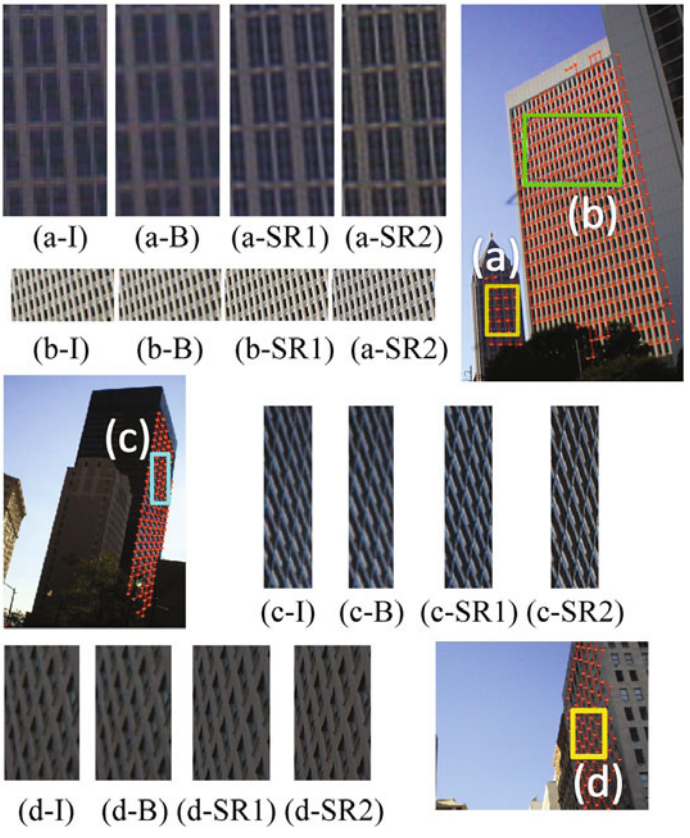
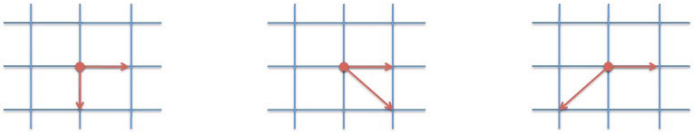


Fig. 9. Sample single view SR results are shown. I stands for original input, B stands for bicubic interpolation, SR-1 stands for super-resolution with the exact copy of the texels, SR-2 stands for super-resolution with a local information transfer such as lighting and shadow. (a-d) input selection. (a-B $\sim$ d-B) results of 2$\times$ bicubic interpolation. (a-SR1,2 $\sim$ d-SR1,2) results of 2$\times$ SR.

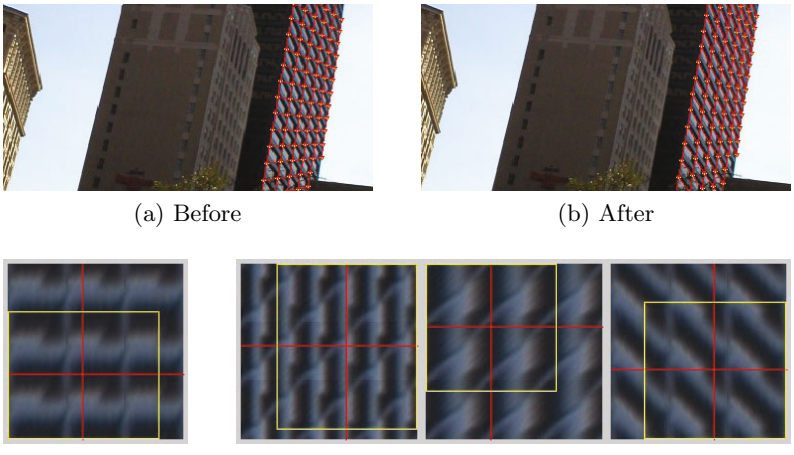
given as $t_1 = p(j, i+1) - p(j, i)$ and $t_2 = p(j+1, i) - p(j, i)$ respectively. The variation of (t_1, t_2) can be given as $t'_1 = p(j, i+1) - p(j, i)$ and $t'_2 = p(j+1, i+1) - p(j, i)$, or $t'_1 = p(j, i+1) - p(j, i)$ and $t'_2 = p(j+1, i-1) - p(j, i)$ as can be seen in Figure 10b or Figure 10c.

We then compute a median texel from quadrilaterals which have been transformed from their 4 observed points in the lattice, $\{p(j, i), p(j, i) + t_1, p(j, i) + t_1 + t_2, p(j, i) + t_2\}$, to rectified points, $\{(1, 1), (w, 1), (w, h), (1, h)\}$ where h and w are the height and width of rectified texels (both set as 50 pixels). Then we tile nine copies of the computed median texel in a 3 by 3 grid to form a small regular lattice pattern and attempt to find the two reflection axes. We only search through x and y directions near the center of the rectified median texel. This is sufficient and necessary because, if reflection axes exist, they must be parallel to the (t_1, t_2) vector pair.



(a) (b) (c)

Fig. 10. (a) The current (t_1, t_2) vector given as $t_1 = p(j, i+1) - p(j, i)$ and $t_2 = p(j+1, i) - p(j, i)$ (b) The variation of (t_1, t_2) vector given as $t_1' = p(j, i+1) - p(j, i)$ and $t_2' = p(j+1, i+1) - p(j, i)$ (c) The variation of (t_1, t_2) vector given as $t_1' = p(j, i+1) - p(j, i)$ and $t_2' = p(j+1, i-1) - p(j, i)$



(a) Before (b) After

(c) Best Chosen (d) Candidate 1 (e) Candidate 2 (f) Candidate 3

Fig. 11. Sample results of correction of (t_1, t_2) vector pair using reflection axes analysis. (a) before (b) after (c) best texel shape (d-f) candidate texel shapes.

We repeat this procedure for all the enumerated (t_1, t_2) vector pairs and seek reflection axes. The sum of the absolute difference between the median texel and the flipped median texel is computed and we select the (t_1, t_2) vector pair that generates the minimum sum as the best pair. Sample results are shown in Figure 11 (c,d,e,f). As can be seen in Figure 11, the analysis is successful in aligning (t_1, t_2) to the vertical and horizontal edges of the building facade.

Computation of Frontal View: Collins and Beveridge [26] showed that when the vanishing points of a 3D plane projected onto an image and the angular field of view of the camera are known, a 3D rotation matrix can be used to relate observed image locations on the plane to the image coordinates they would have if the plane were rotated to face the camera (having a normal vector pointing directly along the camera view direction). Their formulation shows that if the vanishing line of the plane is given by the formula $ax+by+c = 0$, then the normal to that plane, in camera coordinates, is $n = (a, b, c)/\|(a, b, c)\|$. The matrix that

will perform the projective transform simulating the desired 3D rotation is, in homogeneous coordinates,

$$k_i \begin{bmatrix} x_i' \\ y_i' \\ f \end{bmatrix} = \begin{bmatrix} E & F & a \\ F & G & b \\ -a & -b & c \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ f \end{bmatrix} \tag{3}$$

where

$$E = \frac{a^2c + b^2}{a^2 + b^2}, F = \frac{ab(c-1)}{a^2 + b^2}, G = \frac{a^2 + b^2c}{a^2 + b^2} \tag{4}$$

where f is the focal length given by $f = \frac{w}{2tan(FOV/2)}$ where w is the image width and FOV is the camera's angular field of view.

A perspectively distorted lattice that has been identified by our method will converge to two vanishing points, one in each direction of 2D repetition. We can calculate the vanishing points for a lattice covering the facade of a building, then calculate the line connecting the vanishing points in the form $ax + by + c = 0$. From that equation, the values (a, b, c) are the normal vector to the plane in camera coordinates [26]. We use these values to draw the normal vectors to building facades in Fig. 12.

(a) (b)

Fig. 12. This figure shows the computation of surface normals from a lattice detected on a building facade. The blue arrow indicates the surface normal of the building.

We cannot perform the projective transform that would simulate bringing the building facade into a frontal view without knowing the angular field of view of the camera. However, we assume that the two directions of repetition on a building facade are orthogonal in a frontal view. If an incorrect field of view were assumed and used to bring the lattice into a frontal view, the two directions generating the lattice would not be orthogonal. Specifically, an incorrect field of view used to generate the frontal view will induce a scaling along the direction of the facade normal in image coordinates. In our supplemental material, we show that a simple search routine can quickly converge upon the one unique value for angular field of view that can be used to bring a lattice into a frontal view while preserving the orthogonality of the directions of 2D repetition. We show the computed size and shape of the lattice and texels for three images in Fig. 12.

This is a powerful application of our method because computation of building facade normals can be used for 3D reconstruction and geotagging. The calculation of a frontal view of a building facade also can enable extraction of the building appearance as a 2D texture, and can be useful for building recognition where only the frontal appearance of a building is known.

4 Conclusion

A novel 2D translation-symmetry-based method of perceptual grouping is presented that shows superior performance in terms of detecting single and multiple lattices in an image over the state-of-the-art algorithm. Perceptual grouping is possible when mid-level information of scene structures is successfully obtained. Also, we have demonstrated that the detected lattice structure can be used for single view super-resolution as well as for 3D orientation estimation in urban scenes. We plan to extend this work on single view 3D urban scene reconstruction and apply mid-level visual features for object categorization.

Acknowledgement. This work is supported in part by an NSF grant IIS-0729363 and a Google Research Award to Dr. Liu.

References

1. Gardner, M.: The new ambidextrous universe: symmetry and asymmetry, from Mirrow reflections to superstrings. W.H. Freeman and Company, New York (1979)
2. Conners, R., Ng., C.: Developing a quantitative model of human preattentive vision, vol. 19. SMC (1989)
3. Weyl, H.: Symmetry. Princeton University Press, Princeton (1952)
4. Grunbaum, B., Shephard, G.: Tilings and Patterns. W.H. Freeman and Company, New York (1987)
5. Liu, Y., Hel-Or, H., Kaplan, C.S., Van Gool, L.: Computational symmetry in computer vision and computer graphics. Foundations and Trends in Computer Graphics and Vision 5, 1–156 (2010)
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. Hays, J., Leordeanu, M., Efros, 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)
8. 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)
9. Park, M., Brocklehurst, K., Collins, R., Liu, Y.: Deformed Lattice Detection in Real-World Images Using Mean-Shift Belief Propagation. IEEE Transactions on Pattern Analysis and Machine Intelligence 31, 1804–1816 (2009)
10. 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)

11. Canada, B.A., Thomas, G.K., Cheng, K.C., Wang, J.Z., Liu, Y.: Automatic lattice detection in near-regular histology array images. In: Proceedings of the IEEE International Conference on Image Processing (2008)
12. Mitra, N.J., Guibas, L., Pauly, M.: Partial and approximate symmetry detection for 3d geometry. ACM Transactions on Graphics 25, 560–568 (2006)
13. Schaffalitzky, F., Zisserman, A.: Geometric grouping of repeated elements within images. In: Forsyth, D., Mundy, J.L., Di Gesú, V., Cipolla, R. (eds.) Shape, Contour, and Grouping 1999. LNCS, vol. 1681, pp. 165–181. Springer, Heidelberg (1999)
14. Korah, T., Rasmussen, C.: Analysis of building textures for reconstructing partially occluded facades. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 359–372. Springer, Heidelberg (2008)
15. Schindler, G., Krishnamurthy, P., Lublinerman, R., Liu, Y., Dellaert, F.: Detecting and Matching Repeated Patterns for Automatic Geo-tagging in Urban Environments. In: Computer Vision and Pattern Recognition, pp. 1–8 (2008)
16. Hamey, L.G.O., Kanade, T.: Computer analysis of regular repetitive textures. In: Proceedings of a Workshop on Image Understanding Workshop. Morgan Kaufmann Publishers Inc., San Francisco (1989)
17. 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)
18. 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)
19. Liu, Y., Tsin, Y., Lin, W.C.: The Promise and Perils of Near-Regular Texture. International Journal of Computer Vision 62, 145–159 (2005)
20. Tsin, Y., Liu, Y., Ramesh, V.: Texture replacement in real images. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 539–544 (2001)
21. Liu, Y., Belkina, T., Hays, J., Lublinerman, R.: Image De-fencing. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
22. Shi, J., Tomasi, C.: Good features to track. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 593–600 (1994)
23. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: BMVC (2002)
24. 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)
25. Glasner, D., Bagon, S., Irani, M.: Super-resolution from a single image. In: ICCV (2009)
26. Collins, R.T., Beveridge, J.R.: Matching perspective views of coplanar structures using projective unwarping and similarity matching. In: Proc. Int. Conf. of Computer Vision and Pattern Recognition, CVPR, pp. 240–245 (1994)

Towards Hypothesis Testing and Lossy Minimum Description Length: A Unified Segmentation Framework

Mingyang Jiang, Chunxiao Li, Jufu Feng, and Liwei Wang

Key Laboratory of Machine Perception (Peking University),
MOE, Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, P.R. China
{jiangmy,licx,fjf,wanglw}@cis.pku.edu.cn

Abstract. We propose a novel algorithm for unsupervised segmentation of images based on statistical hypothesis testing. We model the distribution of the image texture features as a mixture of Gaussian distributions so that multi-normal population hypothesis test is used as a similarity measure between region features. Our algorithm iteratively merges adjacent regions that are "most similar", until all pairs of adjacent regions are sufficiently "dissimilar". Standing on a higher level, we give a hypothesis testing segmentation framework (HT), which allows different definitions of merging criterion and termination condition. Further more, we derive an interesting connection between HT framework and previous lossy minimum description length (LMDL) segmentation. We prove that under specific merging criterion and termination condition, LMDL can be unified as a special case under HT framework. This theoretical result also gives novel insights and improvements on LMDL based algorithms. We conduct experiments on the Berkeley Segmentation Dataset, and our algorithm achieves superior results compared to other popular methods including LMDL based algorithms.

1 Introduction

Image segmentation, the task of partitioning an image into regions with homogeneous texture, is a crucial first step for high-level image understanding. A good segmentation can significantly reduce the complexity of many visual tasks such as object recognition and scene understanding.

In the literature, many models and principles that can lead to good segmentation have been proposed. Traditional clustering algorithms aim at extracting the statistical characteristics of the region data, such as k-means and Mean Shift [1]. NCuts [2] [3] and F&H [4] formulate the segmentation as a graph-cut problem, while several approaches such as [5] aim at combining the cues of homogeneous color or texture with contours in the segmentation process. Because of the huge diversity of definitions of "optimal segmentation", some recent work such as [6] focused on giving a unified evaluation procedure addressing the problem "what is a good segmentation".

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 343–354, 2011.

More recently, an objective metric based on the notion of *lossy minimum description length* (LMDL) has been proposed for evaluating segmentation of images [7] [8] [9] [10]. This metric is built on the definition of an "optimal segmentation", which is the one that minimizes the number of bits needed to code the segmented data, subject to a given distortion. According to the previous LMDL based algorithms, this objective has been shown to be highly consistent with human segmentation of images. Preliminary success of this approach leads to the following important question: why such objective metric fits well for human understanding of images? It has also been noticed that how to choose the distortion parameter is still a main difficulty in LMDL based approaches.

From another point of view, segmentation is widely accepted as an inference problem, i.e., what caused the observed data. The image data can be characterized by sample points from complicated mixed distribution. It is widely accepted that a good segmentation should group image pixels into regions whose statistical characteristics(e.g. color or texture) are homogeneous. Also, in a good segmentation, segments should be statistically different, i.e., they are corresponding to significantly different distributions. In this view, segmentation can be viewed as a *hypothesis testing* problem, which tests the equality of distributions. Statistically identical or similar regions should be merged, otherwise split. There are simple choices of hypothesis test such as Fisher's test and homogeneous ML test [11]. However, such tests can not distinguish between two distributions with the same means but different variances.

Paper Contributions. In this paper, we propose a simple yet effective algorithm for segmentation of images via hypothesis testing. Based on the observation that a homogeneously textured region of a natural image can be well modeled by a Gaussian distribution, we model segmentation as a multi-normal population hypothesis test problem, which tests the equality of normal population means and variances at the same time. A generalized hypothesis testing (HT) framework is then proposed. The main advantages of the HT framework and the specific contributions of this paper are as follows:

1. HT is a general framework that allows different criteria embedded to yield good segmentations. Under this framework, we propose a specific segmentation algorithm, which is comparable or even better than the best segmentation algorithms.

2. We have proved an interesting result that under specific merging criterion and termination condition, the previous lossy minimum description length (LMDL) segmentation can be unified into our HT framework as a special case. This theoretical contribution reveals the statistical base of LMDL criterion.

2 Segmentation by Hypothesis Testing

2.1 Statistical Model of Image Segmentation

We start by introducing the hypothesis testing model of segmentation. For a given image I, we denote the feature vector as $V = (v_1, v_2, ..., v_m) \in \mathbb{R}_{p\times m}$,

in which m is the number of pixels and p is the feature dimension. The goal of image segmentation can be viewed as grouping the image pixels into regions with homogeneous feature properties which will hopefully correspond to objects or object parts. A region R is considered to be homogeneous if its feature values are consistent with having been generated by a particular distribution $P(I \mid \Theta)$, where Θ are the parameters of the distribution. In a good segmentation, different region should correspond to statistically different distributions. Consequently, segmentation can be viewed as a *hypothesis testing* problem, i.e., testing the equality of the corresponding distributions that generate the image region data.

We focus on the case that data are drawn from multivariate Gaussian distribution. In [10], it has been carefully investigated that a homogeneously textured region of a natural image can be well (not exactly) modeled by a Gaussian distribution. We denote the feature vector in each region R_i as V_i, such that $\bigcup_{i=1}^{M} V_i = V$. For the Gaussian case, each V_i is the observed data from a multi-normal distribution $P(I|\mu_i, \Sigma_i)$, in which μ_i and Σ_i are the mean and variance of multi-normal distribution. In a bottom-up process, the merging of two regions can be viewed as a multi-normal population hypothesis test problem [12] [13]:

$$\begin{aligned} H_0 &: \mu_1 = \mu_2 \; and \; \Sigma_1 = \Sigma_2 \\ H_1 &: \mu_1 \neq \mu_2 \; or \quad \Sigma_1 \neq \Sigma_2 \end{aligned} \tag{1}$$

Under certain significance level, if null hypothesis H_0 is accepted, these two regions R_1 and R_2 are statistically "similar" so that can be merged.

2.2 Multi-normal Population Hypothesis Test

We then give the likelihood ratio of the hypothesis test specified in (1).

Suppose each population has m_i observed sample points denoted as v_{ij}, $i = 1, 2; j = 1, 2, ..., m_i$. Note that each sample point v_{ij} has dimension p. Denote the sample mean vector for each population as $\overline{V_i} = \frac{1}{m_i} \sum_{j=1}^{m_i} v_{ij}$, $i = 1, 2$; The total mean vector of these two populations is $\overline{V} = \frac{1}{m} \sum_{i=1}^{2} \sum_{j=1}^{m_i} v_{ij}, m = \sum_{i=1}^{2} m_i$. The scatter matrix for each population is $A_i = \sum_{j=1}^{m_i} (v_{ij} - \overline{V_i})(v_{ij} - \overline{V_i})^T$, $i = 1, 2$; And the total scatter matrix $T = \sum_{i=1}^{2} \sum_{j=1}^{m_i} (v_{ij} - \overline{V})(v_{ij} - \overline{V})^T$.

The likelihood ratio for the null hypothesis H_0 has the following form [12] [13]:

$$\lambda = \frac{\prod_{i=1}^{2} |A_i|^{\frac{m_i}{2}}}{|T|^{\frac{m}{2}}} \cdot \frac{m^{\frac{mp}{2}}}{\prod_{i=1}^{2} m_i^{\frac{m_i p}{2}}} \tag{2}$$

If we have a large number of sample points m, the likelihood ratio λ has the following approximately distribution when H_0 is true: $-2(1-b)\ln\lambda \sim \chi^2(f)$, in which $f = \frac{1}{2}p(p+3), b = (\sum_{i=1}^{2} \frac{1}{m_i - 1} - \frac{1}{m-2})(\frac{2p^2+3p-1}{6(p+3)}) - \frac{p}{(m-2)(p+3)}$.

Under given significance level α, we have $P(\lambda < \lambda_\alpha) = \alpha$ when H_0 is true. This means in hypothesis testing, the probability that H_0 is wrongly rejected is at most α. Based on the likelihood ratio method, when the calculated value of λ is smaller than λ_α, the null hypothesis H_0 should be reject, otherwise accept. Consequently, $\{\lambda < \lambda_\alpha\}$ induces the rejection range of H_0. If the likelihood ratio falls into the rejection range, the corresponding two regions are statistically "dissimilar".

2.3 Segmentation by Hypothesis Testing

The above hypothesis test gives a measure of "how similar two regions look like", from statistical perspective. Before presenting our hypothesis testing segmentation algorithm, another two essential components have to be made clear: merging criterion and termination condition, i.e., which two regions are chosen to be merged in each bottom-up iteration and how to terminate the algorithm.

Merging Criterion. Denote the merging criterion as MC. A reasonable choice is, in the group of pairs of regions that the corresponding null hypothesis H_0s are accepted, the pair of regions that "most likely" to be the same should be merged. In the likelihood ratio mehod, a large value of λ indicates that the corresponding two populations have a large probability to be the same. From this perspective, we merge the two regions that the corresponding H_0 has maximum λ: $MC = \{\max \lambda\}$.

Termination Condition. Denote the termination condition as TC. One choice is to use the threshold λ_α induced by a given significance level α. If there isn't any pair of regions that the corresponding likelihood ratio is larger than λ_α, the algorithm will terminate. We searched α from 0.00001 to 0.5 and found that α should be very small (less than 0.0001) to achieve good segmentations in the Berkeley Segmentation Dataset. This result implies that segmentation of natural images may require very small type I error (The probability that H_0 is wrongly rejected) to derive segmentations that are consistent with human perception. Although this TC is feasible, the choice of α is difficult for the user because a very small change of it will make significantly different segmentations.

In our algorithm, instead, we use a direct way to define the TC. Since we use hypothesis test to merge the pairs of "similar" regions, a natural idea is that TC should be that feature distributions in adjacent regions must be sufficiently dissimilar. In statistics, the Mallows distance is a commonly used metric between two distributions. For two Gaussian distributions $N(\mu_1, \Sigma_1)$ and $N(\mu_2, \Sigma_2)$, the Mallows distance has the following expression [14]:

$$d_M^2 = (\mu_1 - \mu_2)^T(\mu_1 - \mu_2) + tr(\Sigma_1 + \Sigma_2 - 2(\Sigma_1\Sigma_2)^{\frac{1}{2}}) \tag{3}$$

For a given segmentation, we calculate the Mallows distance between all pairs of adjacent segments. The termination condition is defined to be that the minimal d_M is larger than a preselected threshold θ. That is: $TC = \{min\ d_M > \theta\}$. While

in essence, this TC is the same as using test level, both of which can be viewed as the statistical measures of dissimilarity between adjacent regions.

Algorithm. After defining MC and TC, the hypothesis testing segmentation algorithm is summarized as below:

Algorithm 1: Hypothesis Testing

1. **Input:** image data $V = (v_1, v_2, ..., v_m) \in \mathbb{R}_{p\times m}$ and parameter $\theta > 0$;
2. **Initalize:** sets (regions): $R := \{R_i = \{v\} \,|v \in V\}$;
3. **do**

   apply multi-normal hypothesis test on each pair of adjacent regions in R;

   choose distinct sets R_1, R_2 such that the corresponding H_0 has maximum value of λ; (MC is satisfied)

   merge R_1, R_2: $R := (R \backslash \{R_1, R_2\}) \cup (R_1 \cup R_2)$;

   While $|R| > 1$ and $\{min\ d_M \leq \theta\}$; (TC is not satisfied)
4. **Output:** R.

Since hypothesis test requires a certain number of sample points, it is better to obtain the initial regions R by an over-segmentation procedure. In our experiments, we use the publicly available over-segmentation code [15] with parameter $N_{sp} = 200$.

Hypothesis Testing Segmentation Framework. In fact, different definitions of MC and TC induce a different segmentation algorithm. Based on the use of hypothesis testing as the region similarity measure, we can build a segmentation framework, which allows different definition of MC and TC. In the next section, we will prove that segmentation by lossy minimum description length (LMDL) [7] [10] [8] can be unified into hypothesis testing framework as a special case, under specific MC and TC.

3 Segmentation by Hypothesis Testing and Lossy Data Coding: A Unified Approach

In this section, we will first modify the likelihood ratio by considering noise, and then, derive the connection between LMDL and HT framework.

3.1 Noise Modified Likelihood Ratio

In statistical point of view, the image data may have outliers or noise. We assume that images are perturbed with independent additional noise n_w: $E(n_w) = 0, var(n_w) = \Sigma_w$. This noise can either be viewed as noise in the original image, or the perturbation caused from data filtering, quantization, down-sampling or lossy compression. Then the noise-perturbed image region vector V_i can be viewed as having been generated by the noise-mixed distribution: $P(I|\mu_i^*, \Sigma_i^*)$, in which $\mu_i^* = \mu_i + E(n_w) = \mu_i, \Sigma_i^* = \Sigma_i + \Sigma_w$. Here μ_i and Σ_i are the mean and variance of original normal distribution corresponding to V_i without noise.

Given finite samples, Σ_i can be estimated as: $\Sigma_i = \frac{A_i}{m_i}, i = 1, 2$. Denote Σ as the total sample variance matrix for the two normal populations without noise :$\Sigma = \frac{T}{m}$. For mathematical tractability, here we use the biased variance. Then we get the following theorem:

Theorem 1. *The noise-modified likelihood ratio for the null hypothesis H_0 is:*

$$\lambda^* = \frac{\prod_{i=1}^{2} \left|\Sigma_i \Sigma_w^{-1} + I\right|^{\frac{m_i}{2}}}{\left|\Sigma \Sigma_w^{-1} + I\right|^{\frac{m}{2}}} \tag{4}$$

Proof. Rewrite equation (2) considering noise:

$$\lambda^* = \frac{\prod_{i=1}^{2} |A_i + m_i \Sigma_w|^{\frac{m_i}{2}}}{|T + m\Sigma_w|^{\frac{m}{2}}} \cdot \frac{m^{\frac{mp}{2}}}{\prod_{i=1}^{2} m_i^{\frac{m_i p}{2}}} = \frac{\prod_{i=1}^{2} \left|\frac{A_i}{m_i} + \Sigma_w\right|^{\frac{m_i}{2}}}{\left|\frac{T}{m} + \Sigma_w\right|^{\frac{m}{2}}}$$
$$= \frac{\prod_{i=1}^{2} [\left|\Sigma_i \Sigma_w^{-1} + I\right|^{\frac{m_i}{2}} \cdot |\Sigma_w|^{\frac{m_i}{2}}]}{\left|\Sigma \Sigma_w^{-1} + I\right|^{\frac{m}{2}} \cdot |\Sigma_w|^{\frac{m}{2}}} = \frac{\prod_{i=1}^{2} \left|\Sigma_i \Sigma_w^{-1} + I\right|^{\frac{m_i}{2}}}{\left|\Sigma \Sigma_w^{-1} + I\right|^{\frac{m}{2}}}$$

3.2 Lossy Minimum Description Length Segmentation: A Hypothesis Testing Perspective

In [7], Ma et.al proposed an objective segmentation quality measure, which is based on the lossy minimum description length (LMDL) criterion. Given a potentially mixed data set, the "optimal segmentation" is that, over all possible segmentations, minimizes the coding length of the data, subject to a given *distortion*. For data drawn from a mixture of Gaussians, the optimal segmentation can often be found efficiently using an agglomerative clustering approach, called Pairwise Steepest Descent (PSD) algorithm. LMDL criterion has later been applied to image segmentation, known as CTM algorithm [8].

In both approaches, at each iteration, a pair of regions V_1 and V_2 is merged such that the decrease in the coding length due to coding V_1 and V_2 together is maximal. Let $L(V)$ denote the total number of bits needed to encode the region data V. The algorithms terminate when the coding length can no longer be reduced by merging any pair of regions. That is to say, if their exist two regions V_1 and V_2 such that

$$L(V_1 \cup V_2) - L(V_1, V_2) < 0 \quad , \tag{5}$$

PSD and CTM will continue to merge regions.

Consider that the data distribution is Gaussian: $N(\mu, \Sigma)$. For region data $V = (v_1, v_2, ..., v_m) \in \mathbb{R}_{p \times m}$, we assume $m \gg p$ so that we can ignore the asymptotically insignificant terms in the coding length function, which is also done in [7]. Followed by their result, the coding length function for V is:

$$L(V) = \frac{m}{2} \log_2 \det(I + \frac{p}{\varepsilon^2} \Sigma) + \frac{p}{2} \log_2 (1 + \frac{\mu^T \mu}{\varepsilon^2}) \tag{6}$$

Here ε is the distortion.We consider two regions' merging case. If V_1, V_2 are coded separately, each V_i has m_i sample points, m is total number of sample points: $m = \sum_{l=1}^{2} m_i$, then the total number of bits needed is [7]:

$$L(V_1, V_2) = \sum_{i=1}^{2} [L(V_i) + m_i(-\log_2(m_i/m))] \tag{7}$$

Then we have the following proposition:

Proposition 1. *When the noise variance Σ_w satisfies $\Sigma_w = (\varepsilon^2/p)I$, then the merging condition in LMDL segmentation specified in (5) is equivalent to*

$$\lambda^* > \frac{(1 + \frac{\mu^T \mu}{\varepsilon^2})^{\frac{p}{2}}}{(\frac{m}{m_1})^{m_1}(\frac{m}{m_2})^{m_2}(1 + \frac{\mu_1^T \mu_1}{\varepsilon^2})^{\frac{p}{2}}(1 + \frac{\mu_2^T \mu_2}{\varepsilon^2})^{\frac{p}{2}}} \quad , \tag{8}$$

in which λ^* is the noise-modified likelihood ratio defined in (4).

Proof. Replace $L(V_1 \cup V_2)$ and $L(V_1, V_2)$ in (5) using (6) and (7), we can get:

$$\frac{\left|I + \frac{\sum_1}{\varepsilon^2/p}\right|^{\frac{m_1}{2}} \left|I + \frac{\sum_2}{\varepsilon^2/p}\right|^{\frac{m_2}{2}}}{\left|I + \frac{\sum}{\varepsilon^2/p}\right|^{\frac{m}{2}}} \cdot \frac{(\frac{m}{m_1})^{m_1}(\frac{m}{m_2})^{m_2}(1 + \frac{\mu_1^T \mu_1}{\varepsilon^2})^{\frac{p}{2}}(1 + \frac{\mu_2^T \mu_2}{\varepsilon^2})^{\frac{p}{2}}}{(1 + \frac{\mu^T \mu}{\varepsilon^2})^{\frac{p}{2}}} > 1 \quad , \tag{9}$$

in which Σ_1, Σ_2 are sample variance matrices of V_1 and V_2. Under given condition $\Sigma_w = (\varepsilon^2/p)I$, the first term in the left side equals to λ^*. Move the second term in the left side to the right side of the above inequation, we can get (8).

Note that ε is a special case of the noise variance Σ_w, because the distortion ε implies that the noise in all feature dimensions are i.i.d distributed, with variance ε^2/p. In real situation, different feature dimensions may have different noise variances and they may be correlated, as in the general case of Σ_w.

We will next show that under specific merging criterion and termination condition, LMDL can be unified into HT framework.

Proposition 2. ***(Merging Criterion)*** *The merging criterion (MC) in LMDL segmentation is:*

$$MC = \{\max[\lambda^* \cdot \frac{(\frac{m}{m_1})^{m_1}(\frac{m}{m_2})^{m_2}(1 + \frac{\mu_1^T \mu_1}{\varepsilon^2})^{\frac{p}{2}}(1 + \frac{\mu_2^T \mu_2}{\varepsilon^2})^{\frac{p}{2}}}{(1 + \frac{\mu^T \mu}{\varepsilon^2})^{\frac{p}{2}}}]\}.$$

Proof. In PSD and CTM, two regions V_1, V_2 are merged if and only if $L(V_1, V_2) - L(V_1 \cup V_2)$ is positive and maximum. This is equivalent to maximizing the left side of (9).

In the special case where the data are i.i.d samples from a zero-mean Gaussian distribution, the above MC can be simplified as:

$$MC = \{\max[\lambda^* \cdot e^{m \sum_{i=1}^{2} (-\frac{m_i}{m} \ln \frac{m_i}{m})}]\}$$

The second term in this simplified MC is closely related to the entropy. It can be understood as a reliability-weight of the likelihood ratio: firstly, if we fix the

sample proportion $\frac{m_1}{m}$ and $\frac{m_2}{m}$, it has a large value when total sample number m is large, which means larger sample number generally makes the hypothesis test more reliable; secondly, if we fix the total sample number m, this term has a maximum value when $m_1 = m_2 = \frac{m}{2}$. This means under the constraint that there are fixed m samples available, the hypothesis test is reliable when sample number in both populations are balanced because both populations can have plenty of sample points for sample mean/variance estimation.

Proposition 3. *(**Termination Condition**) The termination condition (TC) in LMDL segmentation is that for all pairs of adjacent regions, the corresponding* λ^* *satisfy* $\lambda^* \leq \lambda_\alpha$*. The threshold* λ_α *is given by :*

$$\lambda_\alpha = \frac{(1+\frac{\mu^T\mu}{\varepsilon^2})^{\frac{p}{2}}}{(\frac{m}{m_1})^{m_1}(\frac{m}{m_2})^{m_2}(1+\frac{\mu_1^T\mu_1}{\varepsilon^2})^{\frac{p}{2}}(1+\frac{\mu_2^T\mu_2}{\varepsilon^2})^{\frac{p}{2}}}.$$

Proof. From proposition 1, if there is no pair of regions satisfies (8), PSD and CTM will terminate. Let λ_α equals to the right side of (8), then the proposition holds.

Under this TC, the rejection range of H_0 is adaptively determined by λ_α, which is a function of sample number, sample mean and preselected distortion ε. Consequently, the meaning of this TC is that under adaptively determined rejection range, the corresponding null hypothesis $H_0 s$ of all pairs of adjacent regions are rejected, i.e., statistically dissimilar. Briefly speaking, LMDL criterion tries to find an optimal segmentation that each region has a high self-similarity, and different regions are sufficiently dissimilar. The so-called "similarity" here is measured by multi-normal population hypothesis test. Similarly, the recently proposed TBES algorithm [10] can also be viewed as a special case of HT with a boundary penalty term as the reliability-weight of the likelihood ratio.

HT Framework: Insights and Improvements. According to [8], a typical difficulty in LMDL is the choice of distortion, which reveals the noise scale of image.

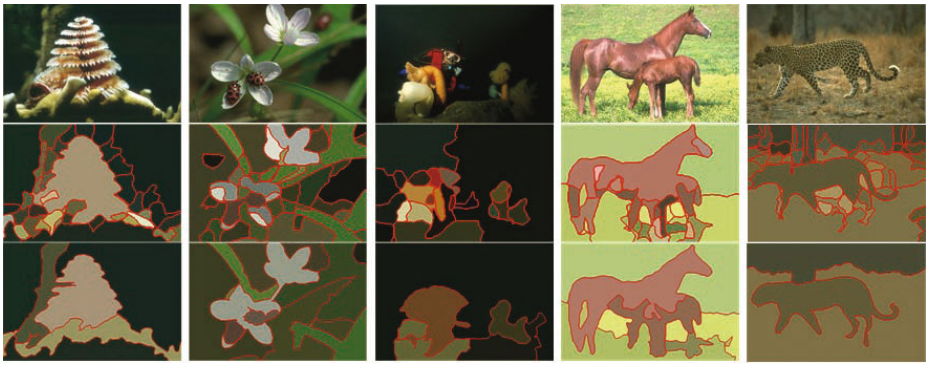


Fig. 1. (color) Result comparison between CTM algorithm and our HT based algorithm. Top row: Original images. Middle row: CTM's results. Bottom row: Our results. Our algorithm better extracts the subject of images from background.

As illustrated in Fig. 1 (row 2): it is often failed to segment out the subject from the background, under the same segmentation scale. This is because CTM uses a fixed ε to code the feature vectors of the entire image, despite the fact that these textures may have different noise variances (e.g., foreground versus background). In a word, distortion is not easy to determine, and is not global consistent even in a single image.

Under HT framework, we require no particular choice of distortion (in the likelihood ratio form in (2)). While the distortion controls the segmentation scale, in our algorithm, instead, we use the Mallows distance threshold θ to play the similar role. It doesn't rely on the distortion, and provides a global consistent measure of the region dissimilarity. As a result, our algorithm successfully extracts the subject of an image from the background (row 3, Fig. 1).

4 Experiments

We conduct extensive experiments to validate the performance of Algorithm 1 on the Berkeley Segmentation Dataset (BSD) [16], which consists of 300 natural images and each of them has been manually segmented by a number of different subjects. We will first describe the feature construction, and then show both qualitative and quantitative results.

4.1 Feature Construction

As shown in Fig. 2, given an image in RGB format, we convert it to the $L*a*b$ color space, which has been investigated in [9] that such color space better facilitates representing texture via mixture of Gaussians. In order to capture the variation of a local texture, we directly apply the 7×7 cut-off window around each pixel. Since the likelihood ratio (2) is uniquely determined by the sample mean and scatter matrix of regions, to estimate them empirically, we need to exclude the windows that cross the boundary of region R. Such windows contains pixels from the adjacent regions, which can not be well modeled by a single Gausssian distribution.

In the next step, we stack the color values inside the window into a vector form. Each window is smoothed by convolving with a 2D Gaussian kernel before stacking. Finally, for the ease of computation, we project the feature vectors into an 8-dimensional space using PCA. The whole procedure and pre-processing are similar as in [10]. From computational point of view, when calculating the likelihood ratio in (2), one can add a relatively small positive number to the diagonal elements of each scatter matrix to ensure that the scatter matrices are positive definite.

4.2 Verification

Qualitative Verification. We first verify the segmentation results on the BSD visually. Since our algorithm relies on the choice of dissimilarity threshold θ, a reasonable result is that smaller θ tends to oversegment images, while larger θ

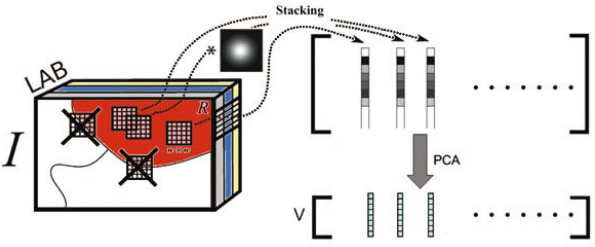


Fig. 2. Feature construction. The 7×7 windows around each pixel on the $L * a * b$ color space are convoluted with 2D Gaussian kernel, stacked into a one column vector and then use PCA.

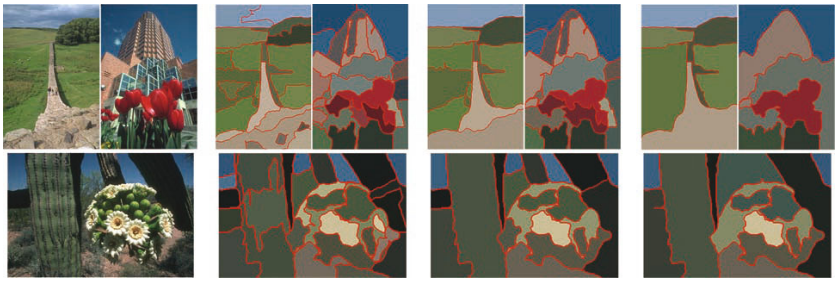


Fig. 3. (color) Results of Hypothesis Testing (HT) segmentation under different θ. Left: Originals. Middle left: $\theta = 0.30$. Middle right: $\theta = 0.45$. Right: $\theta = 0.60$.

tends to undersegment images. Fig. 3 shows the results under $\theta = 0.30, 0.45$ and 0.60. Fig. 4 illustrates more representative segmentation results.

Quantitative Verification. To provide a basis of comparison for the performance of the HT algorithm, we make use of five unsupervised algorithms that have been made *publicly available*: NCuts [2], F&H [4], mean-shift (MS) [1], CTM [8] and TBES [10]. To obtain quantitative evaluation of the performance between our algorithm and the ground truth segmentations in the BSD, we use two widely used quantitative measures: the Probabilistic Rand Index (PRI) [17] and the Variation of Information (VoI) [18]. For brevity, we refer the reader to the stated references for the definition of each index.

The performance of these five methods and that of human's, based on PRI and VOI measures, were obtained via personal communication with the authors of [10]. The user-defined parameters of these methods have been tuned to achieve the best overall tradeoff between PRI and VoI. In particular, we report our results with $\theta = 0.60$.

Note that in Table 1, our method achieves the best result on PRI and the second best result on VoI. It is perhaps not surprising that TBES achieves better VoI since TBES uses additional boundary penalty term to penalize non-smooth contours, while in essence, it can be unified as a special case of HT framework. We

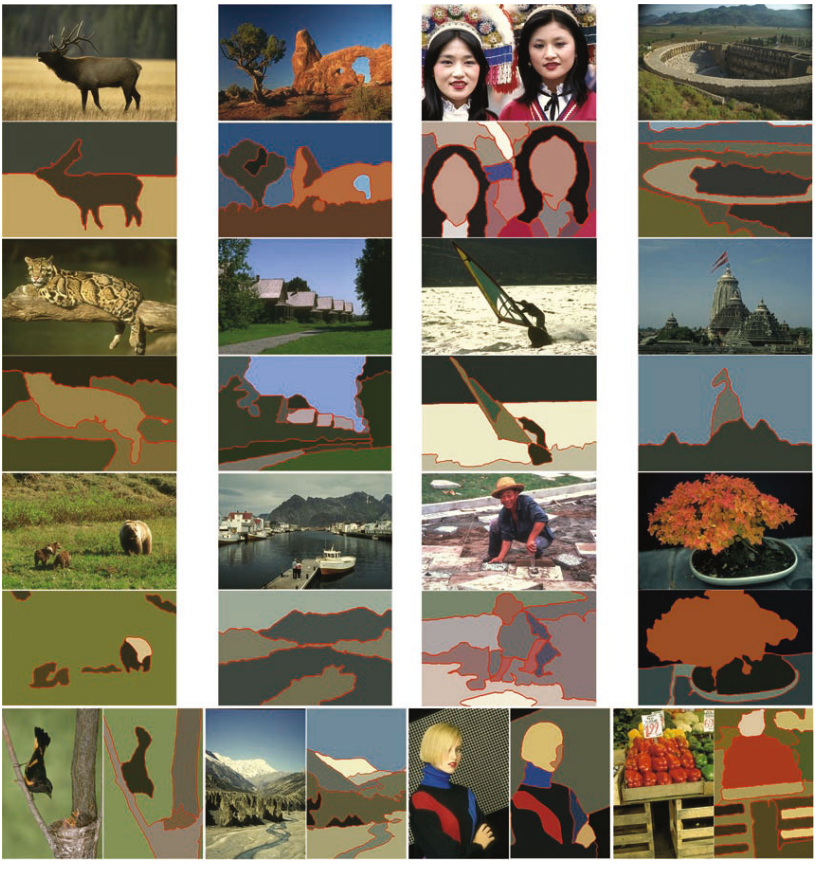


Fig. 4. (color) Qualitative results of our algorithm on the BSD. Each region in the segmented image is colored by its mean color.

Table 1. Quantitative comparison on the BSD. Boldface indicates the best results.

Index/Method	Human	HT	MS	FH	NCuts	CTM	TBES
PRI (Higher is better)	0.868	**0.792**	0.772	0.770	0.742	0.742	0.787
VoI (Lower is better)	1.163	1.897	2.203	2.844	2.651	2.002	**1.824**

also found that, if we could choose θ to optimize the PRI, the average PRI would become 0.804, while similar optimization would bring the VoI down to 1.692.

5 Conclusion

In this work, we have proposed a hypothesis testing segmentation framework which tests population means and variances at the same time.We have proved

that the lossy minimum description length segmentation can be unified into our framework as a special case. This result has found the statistical prototype of LMDL, and gives novel insights and improvements over LMDL based algorithms. Our future direction is to extend this work to non-Gaussian case.

Acknowledgement. This work was supported by NBRPC(2011CB302400), NSFC(60635030), NSFC(61075003) and NSFC(60775005).

References

1. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. PAMI 24, 603–619 (2002)
2. Shi, J., Malik, J.: Normalized cuts and image segmentation. PAMI 22, 888–905 (2000)
3. Kim, T.H., Lee, K.M., Lee, S.U.: Learning full pairwise affinities for spectral segmentation. In: CVPR (2010)
4. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. IJCV 59, 167–181 (2004)
5. Zhu, S.C., Tu, Z.W.: Image segmentation by data-driven markov chain monte carlo. PAMI II, 131–138 (2002)
6. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: From contours to regions: An empirical evaluation. In: CVPR, pp. 2294–2301 (2009)
7. Ma, Y., Derksen, H., Hong, W., Wright, J.: Segmentation of multivariate mixed data via lossy data coding and compression. PAMI 29, 1546–1562 (2007)
8. Yang, A.Y., Wright, J., Ma, Y., Sastry, S.S.: Unsupervised segmentation of natural images via lossy data compression. CVIU 110, 212–225 (2008)
9. Rao, S.R., Mobahi, H., Yang, A.Y., Sastry, S.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)
10. Mobahi, H., Rao, S.R., Yang, A.Y., Sastry, S.S., Ma, Y.: Segmentation of natural images by texture and boundary compression. In: arXiv:1006.3679v1 (2010)
11. Haris, K., Efstratiadis, S.N., Maglaveras, N., Katsaggelos, A.K.: Hybrid image segmentation using watersheds and fast region merging. IEEE Transactions on Image Processing 7, 1684–1699 (1998)
12. Hsieh, H.K.: On asymptotic optimality of likelihood ratio tests for multivariate normal distributions. The Annals of Statistics 7, 592–598 (1979)
13. Perng, S.K., Littell, R.C.: A test of equality of two normal population means and variances. Journal of the American Statistical Association 71, 968–971 (1976)
14. Dowson, D., Landau, B.: The frechet distance between multivariate normal distributions. Journal of Multivariate Analysis 12, 450–455 (1982)
15. Mori, G.: Guiding model search using segmentation. In: ICCV (2005)
16. 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: ICCV, vol. 2, pp. 416–423 (2001)
17. Rand, W.M.: Objective criteria for the evaluation of clustering methods. American Statistical Association Journal 66, 846–850 (1971)
18. Meila, M.: Comparing clusterings: An axiomatic view. In: ICML (2005)

A Convex Image Segmentation: Extending Graph Cuts and Closed-Form Matting

Youngjin Park and Suk I. Yoo

Seoul National University, Kwanak, Seoul, Korea
{yjpark,siyoo}@ailab.snu.ac.kr

Abstract. Image matting and segmentation are two closely related topics that concern extracting the foreground and background of an image. While the methods based on global optimization are popular in both fields, the cost functions and the optimization methods have been developed independently due to the different interests of the fields: graph cuts optimize combinatorial functions yielding hard segments, and closed-form matting minimizes quadratic functions yielding soft matte.

In this paper, we note that these seemingly different costs can be represented in very similar convex forms, and suggest a generalized framework based on convex optimization, which reveals a new insight. For the optimization, a primal-dual interior point method is adopted. Under the new perspective, two novel formulations are presented showing how we can improve the state-of-the-art segmentation and matting methods. We believe that this will pave the way for more sophisticated formulations in the future.

1 Introduction

Estimating the foreground and background of an image is of great importance in computer vision. When the boundary details are not critical, hard segmentation methods are usually employed that assign a label to every pixel. Graph cuts are among the most successful methods in this class. This might be sufficient for some tasks such as recognition, however, an accurate soft matte (foreground opacity) is often required, for instance, in image editing. Recently, the closed-form formulation using *matting Laplacian* [15] has been proven to be very effective for matting [26]. While generating accurate boundaries, however, it usually requires the unknown region to be small unlike graph cuts. This sort of complementary property has triggered the development of algorithms that take advantage of both sides [18]. Our work reveals the link between the two problems, and the better approaches are obtained through generalization. Note that we will consider the two-layer (foreground and background) segmentation and matting only, but multiple layers can be handled in a standard manner [5,23].

Segmentation is one of the most intensively studied topics in computer vision. Among many others, graph cuts methods have gained the popularity due to their capability to infer globally consistent labels incorporating various local cues. While the original formulation is in a combinatorial form [10], it is known that its

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 355–368, 2011.

continuous relaxation results in a convex function. In particular, [2] reformulated graph cuts as an unconstrained l_1 minimization problem, and solved it using the barrier method, a general algorithm for solving convex problems. Similarly, we reformulated the cost function into the following form (Sec. 2.1):

$$J_1(\alpha) = \|K\alpha\|_1 + \|V(\alpha - \hat{\alpha}_1)\|_1 \tag{1}$$

where α is the relaxed soft segmentation labels. The two terms on the right side corresponds to smoothness and data terms.

The problem of finding an accurate matte has gained attention relatively recently. Currently, many of the state-of-the-art methods [12, 17, 18, 22, 27] are derived from the work of [15]. Generalizing the cost function of these methods yields the following expression (Sec. 2.2):

$$J_2(\alpha) = \|L^{1/2}\alpha\|_2^2 + \|W^{1/2}(\alpha - \hat{\alpha}_2)\|_2^2 \tag{2}$$

The similarities of the two expressions are apparent; they are l_1 and l_2 norms of the same form.

We recognize this close relationship between the two. Since they are both convex, the sum of them also yields a convex function allowing us to efficiently find a global minimum. Hence, we can see graph cuts and closed-form matting methods within the convex optimization framework. As a result, we propose a new convex segmentation framework whose cost function is given as

$$J(\alpha) = J_1(\alpha) + J_2(\alpha) = \|A\alpha + b\|_1 + (1/2)\,\|C\alpha + d\|_2^2 \tag{3}$$

In this expression, we further merged the smoothness and the data terms, and simply obtained l_1 and l_2 norms of the linear functions of α. While it is possible to consider other types of convex functions, we focus on this combination since each of them is well studied.

This generalization is of more than just the theoretical interests; in fact, by properly choosing the parameters A, b, C, and d, we can improve both segmentation and matting quality. We will show two particular examples in Sec. 5. For matting, Sec. 5.1 incorporates l_1 data terms, and this makes the method robust to erroneous data estimates. For segmentation, Sec. 5.2 adopts matting Laplacian as smoothness terms, and shows that the shrinking phenomena of graph cuts are mitigated.

One related work is [24]. They have obtained a new segmentation algorithm based on l_∞ norms by generalizing graph cuts and random walker [9]. We further generalize the formulation so that the closed-form matting can be included, and try optimizing joint cost function.

For the optimization, we used a primal-dual interior point method which is a standard technique for convex optimization. Since our problem is often very large including millions of variables and inequality constraints, most general solvers cannot handle it; we implemented new software for minimizing Eq. (3). Exploiting GPU computing technology, we could make it fast enough to be practically used.

Similar formulations and optimization methods often occur in some other fields including sparse signal reconstruction, feature selection, and statistics [13]. For image restoration, [7] also used a primal-dual interior point method minimizing mixed l_1 and l_2 norms. [1] adopted a similar cost function for the feature learning problem.

In summary, this paper gives a new perspective on segmentation and matting. We showed the relationship between graph cuts and closed-form matting. Following the observation, we developed a new method and obtained promising results. We believe that developing new convex objectives is an attractive future research direction.

2 Previous Works

We start by clarifying our notation. The c channel color input image is denoted as $\mathcal{I}$, and $\mathcal{I}_i$ is a $c \times 1$ vector representing colors of the ith pixel. We want to estimate the foreground opacity $\mathbf{0} \preceq \alpha \preceq \mathbf{1}$, a column vector of length n where n is the number of pixels. The curled inequality denotes componentwise inequality. $\mathbf{0}$ and $\mathbf{1}$ are column vectors of 0 and 1, respectively, whose size should be apparent from the context. Also, $\|\cdot\|_i$ represents l_i norm. Thus, $\|x\|_1 = \sum_i |x_i|$ and $\|x\|_2^2 = \sum_i x_i^2$ for a vector x.

2.1 Graph Cuts

Graph cuts methods treat the image as a graph, and divide it by finding a minimum cut of it [10]. The cost function is given as

$$J(\alpha) = \sum_{(i,j)\in N} f_{ij}(\alpha_i, \alpha_j) + \sum_i f_i(\alpha_i) \tag{4}$$

where N is the neighborhood or edge set, and $\alpha_i \in \{0, 1\}$. It consists of two parts: the smoothness terms that are functions of two neighboring pixels, and the data terms encoding local likelihoods. Smoothness terms are designed to prefer similar pixels to be in the same segment, thus, usually defined as a dissimilarity of neighboring pixels.

The minimum cut is usually found by solving its dual problem: max flow problem. It involves the classic Ford-Fulkerson [6,8] or Push-relabel with their specialized improvements.

While the original formulation restricts the labels to the discrete values, its continuous relaxation allows $\alpha \in [0, 1]$ instead. The cost function should also be adapted to be defined on the continuous domain. In fact, it is known that a convex continuous relaxation is possible, if f_{ij} is submodular; that is, if $f_{ij}(0,0) + f_{ij}(1,1) \leq f_{ij}(0,1) + f_{ij}(1,0)$. One possible form is presented in [2], leading to an unconstrained l_1 minimization problem, and it is solved using the barrier method.

Similarly, we reformulate Eq. (4) as the following convex continuous form:

$$J_1(\alpha) = \|K\alpha\|_1 + \|V(\alpha - \hat{\alpha}_1)\|_1 \tag{5}$$

where each row r of K corresponds to an edge $(i, j) \in N$ with its two non-zero elements being defined as

$$K_{r,i} = -K_{r,j} = \left(-f_{ij}(0,0) + f_{ij}(0,1) + f_{ij}(1,0) - f_{ij}(1,1)\right)/2 \tag{6}$$

and the diagonal matrix V and $\hat{\alpha}$ are given as

$$V_{i,i} = |w_i|, \quad \hat{\alpha}_{1i} = \begin{cases} 1 & \text{if } w_i \geq 0, \\ 0 & \text{if } w_i < 0, \end{cases} \tag{7}$$

where w_i is defined as

$$w_i = f_i(1) - f_i(0) - \sum_{j|(i,j)\in N} \frac{f_{ij}(0,0) + f_{ij}(0,1) - f_{ij}(1,0) - f_{ij}(1,1)}{2} \tag{8}$$

It is easy to verify that any minimum of Eq. (4) is also a minimum of Eq. (5).

2.2 Closed-Form Matting

Matting problem is to estimate a foreground and a background of an image along with a opacity for each pixel. Most algorithms typically assume compositing equation:

$$\mathcal{I}_i = \alpha_i \mathcal{F}_i + (1 - \alpha_i)\mathcal{B}_i \tag{9}$$

where $\mathcal{F}_i$ and $\mathcal{B}_i$ are $c \times 1$ vectors representing foreground and background colors of the ith pixel, respectively. Since the problem is ill-posed, usually a trimap indicating definite foreground C_F, background C_B, and unknown region is given. We enforce $\alpha_i = 1$ for $i \in C_F$, and $\alpha_i = 0$ for $i \in C_B$.

Currently, matting-Laplacian-based methods produce the best results. They define a quadratic cost function of α:

$$J(\alpha) = \alpha^T L \alpha \tag{10}$$

where L is a $n \times n$ symmetric matrix referred as the matting Laplacian that we define:

$$L_{i,j} = \sum_{k|i,j\in w_k} \left(\delta_{ij} - \frac{1}{|w_k|}\left(1 + (\mathcal{I}_i - \mu_k)(\Sigma_k + \frac{\epsilon}{|w_k|} I_c)^{-1}(\mathcal{I}_j - \mu_k)\right)\right). \tag{11}$$

where δ_{ij} is Kronecker delta, Σ_k is a $c \times c$ covariance matrix, μ_k is a $c \times 1$ mean vector of the colors in a window w_k, and I_c is the $c \times c$ identity matrix. Typically, $c = 3$ for color images. See [15] for the original derivation.

There is another way of defining matting Laplacian. While the above is based on color line assumption, [22] assumed locally constant color model for an alternative derivation. In both cases, the cost function is in a quadratic form.

Often, quadratic data terms or priors are added [12,17,18,27].

$$J(\alpha) = \alpha^T L\alpha + (\alpha - \hat{\alpha})W(\alpha - \hat{\alpha}) \tag{12}$$

where W is a diagonal matrix penalizing α from deviating from $\hat{\alpha}$.

We may see Eq. (12) as a combination of closed-form matting and random walker segmentation [9, 17, 27]. Similarities of their cost functions allow easy integration; the same optimization method, solving the following sparse linear system, can be applied.

$$(L + W)\alpha = W\hat{\alpha} - b \tag{13}$$

One noticeable shortcoming of matting Laplacian methods is that the solution often includes mid-range values; in reality, the alpha values of 0 and 1 are more likely. Enforcing sparsity prior in quadratic forms are inherently difficult as we will see in Sec. 5.1.

Rewriting Eq. (12) yields the following expression:

$$J_2(\alpha) = (1/2)\|L^{1/2}\alpha\|_2^2 + (1/2)\|W^{1/2}(\alpha - \hat{\alpha}_2)\|_2^2 \tag{14}$$

3 A Convex Segmentation

Many of the best segmentation algorithms can be understood as optimizing a convex function. We refer those as convex segmentation methods. They are particulary interesting since the convexity allows the efficient computation of a global optimum. Among numerous possible convex functions, we suggest a particular form comprising l_1 and l_2 norms.

Formally, given an input image $\mathcal{I}$ with n pixels, we obtain a foreground opacity α_i for each pixel i, by solving the following optimization problem:

$$\begin{aligned} \underset{\alpha}{\text{minimize}} \quad & J(\alpha) = J_1(\alpha) + J_2(\alpha) = \|A\alpha + b\|_1 + (1/2)\,\|C\alpha + d\|_2^2 \\ \text{subject to} \quad & \alpha_i = 1,\ i \in C_F, \\ & \alpha_i = 0,\ i \in C_B, \\ & 0 \le \alpha_i \le 1,\ i = 1 \ldots n \end{aligned} \tag{15}$$

where C_F and C_B are the sets of pixels that are pre-specified as definite foreground and background, respectively. Each component of α is constrained to be in $[0, 1]$.

3.1 Specializations

The convex segmentation problem of Eq. (15) includes graph cuts and closed-form matting methods. If we let $A = \begin{bmatrix} K \\ V \end{bmatrix}$, $b = \begin{bmatrix} \mathbf{0} \\ -V\hat{\alpha}_1 \end{bmatrix}$, $C = \mathbf{0}^T$, and $d = 0$, then Equ. (15) reduces to graph cuts cost of Eq. (5). If we let $A = \mathbf{0}^T$, $b = 0$, $C = \begin{bmatrix} L^{1/2} \\ W^{1/2} \end{bmatrix}$, and $d = \begin{bmatrix} \mathbf{0} \\ -W^{1/2}\hat{\alpha}_2 \end{bmatrix}$, then Eq. (15) reduces to closed-form matting of Eq. (14). Note that since L and W are positive semidefinite, $L^{1/2}$ and $W^{1/2}$ are valid.

3.2 Remarks

In our framework, we have no restriction on A, b, C, and d. This means that we may have the expressive power beyond the simple mixture of graph cuts and closed-form matting. For example, the components of $\hat{\alpha}_1$ are constrained to be either 0 or 1 in graph cuts, but any value is possible in the new form (Fig. 1); Sec. 5.1 shows the usefulness of this extension. Also, supermodular cost functions that graph cuts cannot minimize are allowed in this formulation. In fact, Fig. 1 (a) depicts such a case: $f_{ij}(0,0) + f_{ij}(1,1) > f_{ij}(0,1) + f_{ij}(1,0)$.

The l_1 and l_2 norm minimization problems are well studied in convex optimization. l_1 norm minimization is robust to data noise and usually leads to a sparse solution; l_2 norm minimization is not robust but has stable solution. These properties also apply to our segmentation problem as we will see in Sec. 5.

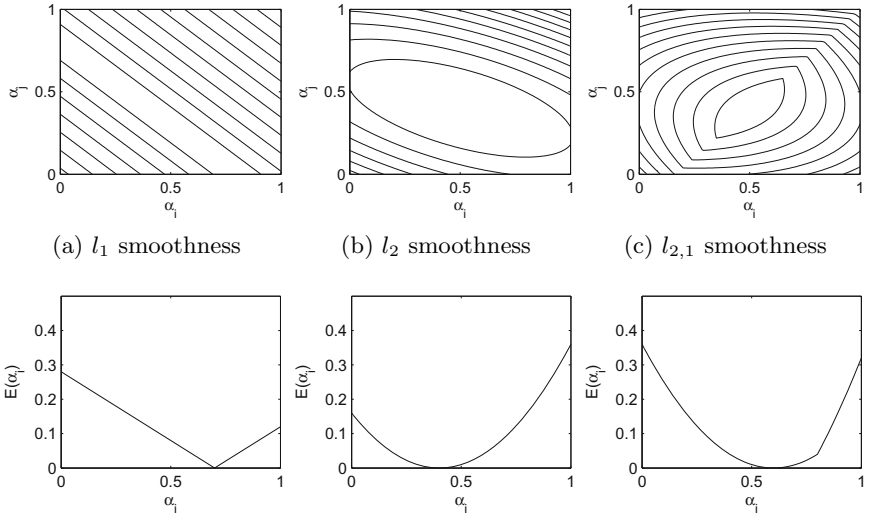


Fig. 1. Possible cost functions: the top row shows the level contours of the cost with respect to two labels, and the bottom row plots the cost as a function of a single label. Our new formulation allows complicated relationships such as (c) and (f) previously impossible.

4 Optimization

This section gives a summary on solving the problem of Eq. (15) using a primal-dual interior point method. Although the derivation here is tailored for the l_1 and l_2 norms, other convex forms might be added harmlessly. We assumed that readers have some knowledge of convex optimization due to the page limit. Refer the text of [3] for deeper understanding.

4.1 Reformulation

First, we reformulate the problem so that it can be better handled. First, we remove the equality constraints by substitution. Though we may incorporate them into data terms using large coefficients as in [12], or may just leave it since the primal-dual interior point method can handle them [3], substitution is a better strategy since it reduces the number of variables significantly when dealing with a large trimap. Substituting $\alpha_i = 1$ for $i \in C_F$ and $\alpha_i = 0$ for $i \in C_B$ yields:

$$\begin{aligned} &\underset{\alpha_U}{\text{minimize}} \quad J(\alpha_U) = \left\| A_{(:,U)}\alpha_U + A_{(:,F)}\mathbf{1} + b \right\|_1 + (1/2)\left\| C_{(:,U)}\alpha_U + C_{(:,F)}\mathbf{1} + d \right\|_1 \\ &\text{subject to } \mathbf{0} \preceq \alpha_U \preceq \mathbf{1} \end{aligned} \tag{16}$$

where $\cdot_{(:,U)}$ and $\cdot_{(:,F)}$ give new matrices only with the columns corresponding unconstrained and foreground pixels, respectively, and α_U is α only with unconstrained pixels. Since $A_{(:,F)}\mathbf{1} + b$ and $C_{(:,F)}\mathbf{1} + d$ are again constant column vectors, this substitution does not change the form of the equation; it just removes some columns and rows. Hence, without loss of generality, we will assume the case with no equality constraints from Eq. (15):

$$\begin{aligned} &\underset{\alpha}{\text{minimize}} \quad J(\alpha) = \|A\alpha + b\|_1 + (1/2)\|C\alpha + d\|_2^2 \\ &\text{subject to} \quad 0 \le \alpha_i \le 1,\ i = 1 \dots n \end{aligned} \tag{17}$$

Second, we introduce an auxiliary variable w to deal with the nondifferentiability of l_1 norms. It is a standard technique, and is also used in [2].

$$\begin{aligned} &\underset{\alpha,w}{\text{minimize}} \quad J(\alpha, w) = \mathbf{1}^T w + (1/2)\|C\alpha + d\|_2^2 \\ &\text{subject to} \quad -w \preceq A\alpha + b \preceq w \\ &\qquad\qquad \mathbf{0} \preceq \alpha \preceq \mathbf{1} \end{aligned} \tag{18}$$

Eq. (18) is equivalent to Eq. (17), but has a differentiable objective with additional inequality constraints. Rewriting once more the inequality constraints gives the final smooth form that we will handle:

$$\begin{aligned} &\underset{\alpha,w}{\text{minimize}} \quad J(\alpha, w) = \mathbf{1}^T w + (1/2)\|C\alpha + d\|_2^2 \\ &\text{subject to} \quad Z\begin{bmatrix}\alpha \\ w\end{bmatrix} + Y \preceq \mathbf{0},\ \text{where } Z = \begin{bmatrix} -A & -I \\ A & -I \\ -I & \mathbf{0} \\ I & \mathbf{0} \end{bmatrix} \text{ and } Y = \begin{bmatrix} -b \\ b \\ \mathbf{0} \\ -\mathbf{1} \end{bmatrix} \end{aligned} \tag{19}$$

4.2 A Primal-Dual Interior Point Method

We start from the Karush-Kuhn-Tucher (KKT) optimality conditions for Eq. (19):

$$\begin{aligned} Z\begin{bmatrix}\alpha^*\\ w^*\end{bmatrix} + Y &\preceq \mathbf{0}, \\ \lambda^* &\succeq \mathbf{0}, \\ \mathrm{diag}(\lambda^*)\left(Z\begin{bmatrix}\alpha*\\ w^*\end{bmatrix} + Y\right) &= \mathbf{0}, \\ \begin{bmatrix}C^TC\alpha^* + C^Td\\ \mathbf{1}\end{bmatrix} + Z^T\lambda^* &= \mathbf{0} \end{aligned} \tag{20}$$

where the column vector λ is a dual variable of length m associated with m inequality constraints. These conditions must be satisfied by any pair of primal and dual optimal points α^*, w^*, and λ^*.

We cannot find such a point analytically, so we resort to a sequential numerical algorithm. We update the current point (α, w, λ) following the primal-dual search direction $(\Delta\alpha, \Delta w, \Delta\lambda)$. Until the convergence, the next point $(\alpha^+, w^+, \lambda^+)$ is obtained:

$$\alpha^+ = \alpha + \tau\Delta\alpha, \quad w^+ = w + \tau\Delta w, \quad \lambda^+ = \lambda + \tau\Delta\lambda \tag{21}$$

where τ is called a search step.

Search directions are obtained by Newton's method applied to a series of modified KKT equations expressed as $r_t(\alpha, w, \lambda) = \mathbf{0}$, where we define: (cf. Eq. (20))

$$r_t(\alpha, w, \lambda) = \begin{bmatrix} \begin{bmatrix}C^TC\alpha + C^Td\\ \mathbf{1}\end{bmatrix} + Z^T\lambda \\ -\mathrm{diag}(\lambda)\left(Z\begin{bmatrix}\alpha\\ w\end{bmatrix} + Y\right) - (1/t)\mathbf{1} \end{bmatrix} \tag{22}$$

The search step τ is decided so that the updated values still satisfy the two inequality constraints of Eq. (20):

$$\tau = \sup\{\tau \in [0,1] \mid Z\begin{bmatrix}\alpha + \tau\Delta\alpha\\ w + \tau\Delta w\end{bmatrix} + Y \preceq \mathbf{0}, \lambda + \tau\Delta\lambda \succeq \mathbf{0}\} \tag{23}$$

After one update, t is recalculated using the surrogate duality gap [3]:

$$t = \mu m/(-(Z\begin{bmatrix}\alpha\\ w\end{bmatrix} + Y)^T\lambda) \tag{24}$$

where μ is a parameter that works well on the order of 10.

Note that as t increases, the modified KKT equation better approximates the equality conditions of Eq. (20), while the inequality conditions are satisfied by Eq. (23). Hence, the solution converges to a global minimum satisfying all the KKT conditions.

4.3 A Newton Step

The Newton step solves the nonlinear equation $r_t(\alpha, w, \lambda) = \mathbf{0}$ for a fixed t by forming Taylor approximation at the current point $x = (\alpha, w, \lambda)$ yielding a search direction $\Delta x = (\Delta\alpha, \Delta w, \Delta\lambda)$:

$$r_t(x + \Delta x) \approx r_t(x) + Dr_t(x)\Delta x = \mathbf{0} \tag{25}$$

In terms of α, w, and λ,

$$\begin{bmatrix} H & Z^T \\ -\mathrm{diag}(\lambda)Z & \mathrm{diag}(-(Z\begin{bmatrix}\alpha\\ w\end{bmatrix} + Y)) \end{bmatrix} \begin{bmatrix}\Delta\alpha\\ \Delta w\\ \Delta\lambda\end{bmatrix} = -r_t(\alpha, w, \lambda) \tag{26}$$

where $H = \begin{bmatrix} C^TC & 0 \\ 0 & 0 \end{bmatrix}$. A block elimination yields:

$$\left[H + Z^T\mathrm{diag}(\lambda)\mathrm{diag}(s)^{-1}Z\right]\begin{bmatrix}\Delta\alpha\\ \Delta w\end{bmatrix} = -\begin{bmatrix} C^TC\alpha \\ \mathbf{1}\end{bmatrix} - (1/t)Z^T\mathrm{diag}(s)^{-1}\mathbf{1} \tag{27}$$

where $s = -(Z\begin{bmatrix}\alpha\\ w\end{bmatrix} + Y)$.

We solve the sparse linear system of Eq. (27) using the conjugate gradient method. Our GPU implementation is influenced by [4]; we also used Jacobian preconditioner. This leaves rooms for further significant speedup using sophisticated preconditioners.

Once the primal search direction $(\Delta\alpha, \Delta w)$ is obtained, the dual search direction $\Delta\lambda$ is given as

$$\Delta\lambda = \mathrm{diag}(s)^{-1}\mathrm{diag}(\lambda)Z\begin{bmatrix}\Delta\alpha\\ \Delta w\end{bmatrix} - \lambda + \mathrm{diag}(s)^{-1}(1/t)\mathbf{1} \tag{28}$$

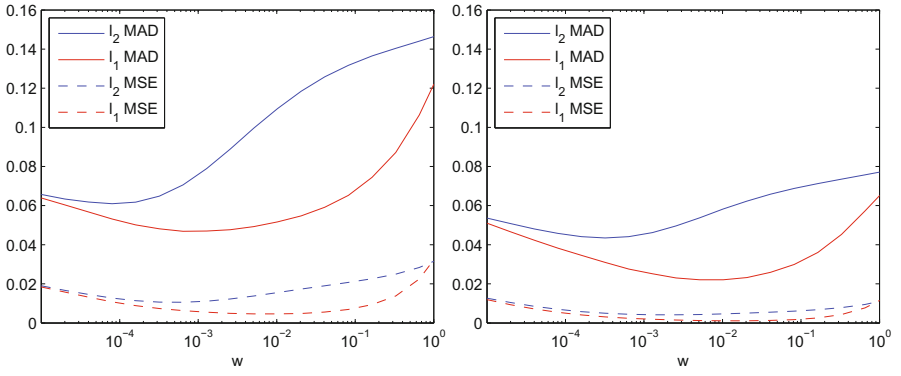
5 Applications

Having defined the new convex form and knowing that we can optimize it, this section shows the examples that actually benefit from that. Note that the cost functions of this section need to be transformed to fit in Eq. (15) before being optimized. This should be easy following Sec. 2. For every experiment in this section, the computation time for each image was less than 10 seconds.

5.1 Matting

Many of the current state-of-the-art matting methods incorporate l_2 data terms based on certain global color models resulting in the following cost function (see Sec. 2.2):

$$J(\alpha) = \alpha^T L\alpha + (\alpha - \hat{\alpha})W(\alpha - \hat{\alpha}) \tag{29}$$



(a) $\epsilon = 0.2, \sigma = 0.2$ (b) $\epsilon = 0.1, \sigma = 0.1$

Fig. 2. Robust data terms for matting: using l_1 data terms almost always give lower errors under the assumption of the estimation model Eq. (31): varying ϵ and σ does not change the general shape of the plot.

where $\hat{\alpha}_i$ is the most likely opacity value for the pixel i. However, often the estimation of $\hat{\alpha}_i$ is erroneous due to imperfect color models, and the final result is easily affected by small amount of outliers. In such cases, it is known that using l_1 norms, instead, gives robust results. Hence, we designed a new formulation:

$$J(\alpha) = \alpha^T L \alpha + \|W(\alpha - \hat{\alpha})\|_1 \tag{30}$$

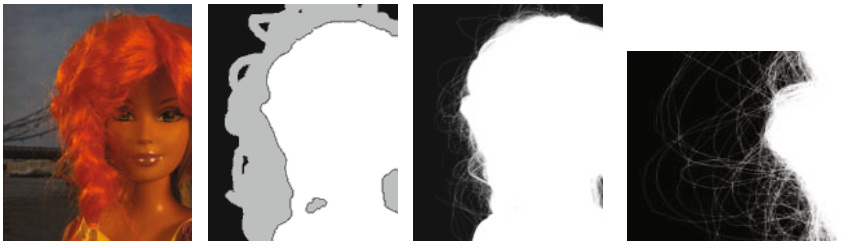
Our expectation is that minimizing Eq. (30) gives more accurate matte than minimizing Eq. (29), and we have experimentally confirmed this. For the fair comparison of the two, we have assumed the following hypothetical estimation model rather than resorting to a particular method:

$$\hat{\alpha}_i \sim \begin{cases} U(0,1) & \text{with probability } \epsilon, \\ \gamma_i + N(0,\sigma^2) & \text{with probability } (1-\epsilon) \end{cases} \tag{31}$$

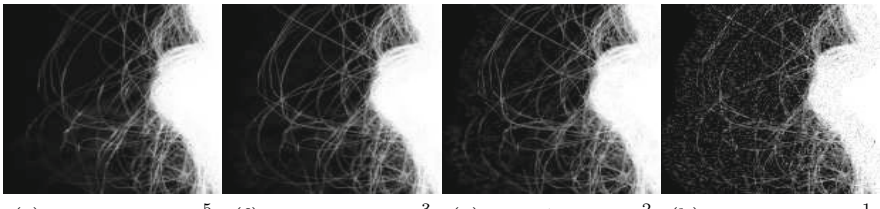
where $U(0,1)$ is a uniform distribution, γ_i is the ground truth opacity, and $N(0,\sigma^2)$ is a Gaussian distribution with variance σ^2; we simulate the measurement of the term $\hat{\alpha}_i$. Hence, the measurement is incorrect with the probability ϵ. Also, W is assumed to be a diagonal matrix whose diagonal elements are all w.

This experiment requires the ground truth matte, so we used the training dataset of [19]. We have measured the error with respect to w, which is the weight of the data term. Fig. 2 shows that using l_1 norm significantly reduces the errors. This is the result averaged over all 27 images.

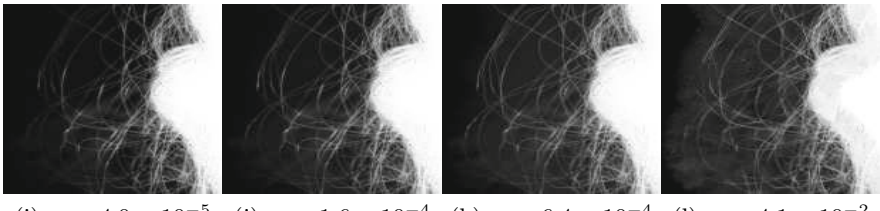
Since the two error measures, mean absolute difference (MAD) and mean squared error (MSE), are sometimes inconsistent with the perceptual quality [19], we also examined the results qualitatively, but we found no inconsistency in this case. Fig. 3 shows the close-up look at one of the results. As expected, we could find the problem of non-sparse solution is relieved when the new l_1 data terms are used.



(a) Input Image (b) Trimap (c) Ground Truth (d) Zoomed-In GT



(e) $w = 8.0 \times 10^{-5}$ (f) $w = 5.1 \times 10^{-3}$ (g) $w = 4.1 \times 10^{-2}$ (h) $w = 3.3 \times 10^{-1}$



(i) $w = 4.0 \times 10^{-5}$ (j) $w = 1.6 \times 10^{-4}$ (k) $w = 6.4 \times 10^{-4}$ (l) $w = 4.1 \times 10^{-2}$

Fig. 3. Qualitative comparison between different data terms for matting. It shows the zoomed-in results for a particular region. The second row: Eq. (30) with l_1 data terms. The third row: Eq. (29) with l_2 data terms. (f) and (j) are the best cases. We can see that the hairs stand out clear in the l_1 case.

5.2 Segmentation

We may also improve the segmentation quality by replacing the smoothness term of graph cuts with the well-defined matting Laplacian term. Our new formulation has the following form:

$$J(\alpha) = \lambda \alpha^T L \alpha + \sum_i f_i(\alpha_i) \tag{32}$$

Of course, since f_i is only defined when $\alpha_i \in \{0, 1\}$, the continuous relaxation (in Sec. 2.1) is required. In this way, we can avoid the heuristic step often involved in defining the smoothness terms. Also, a well known shrinking bias of graph cuts can be mitigated. However, since this results in a soft segmentation, the final thresholding step is required; we used the fixed threshold of 0.5.

We first implemented GrabCut method [21] and tried substituting the matting Laplacian term for the smoothness term. The weighting constant λ is set to 10

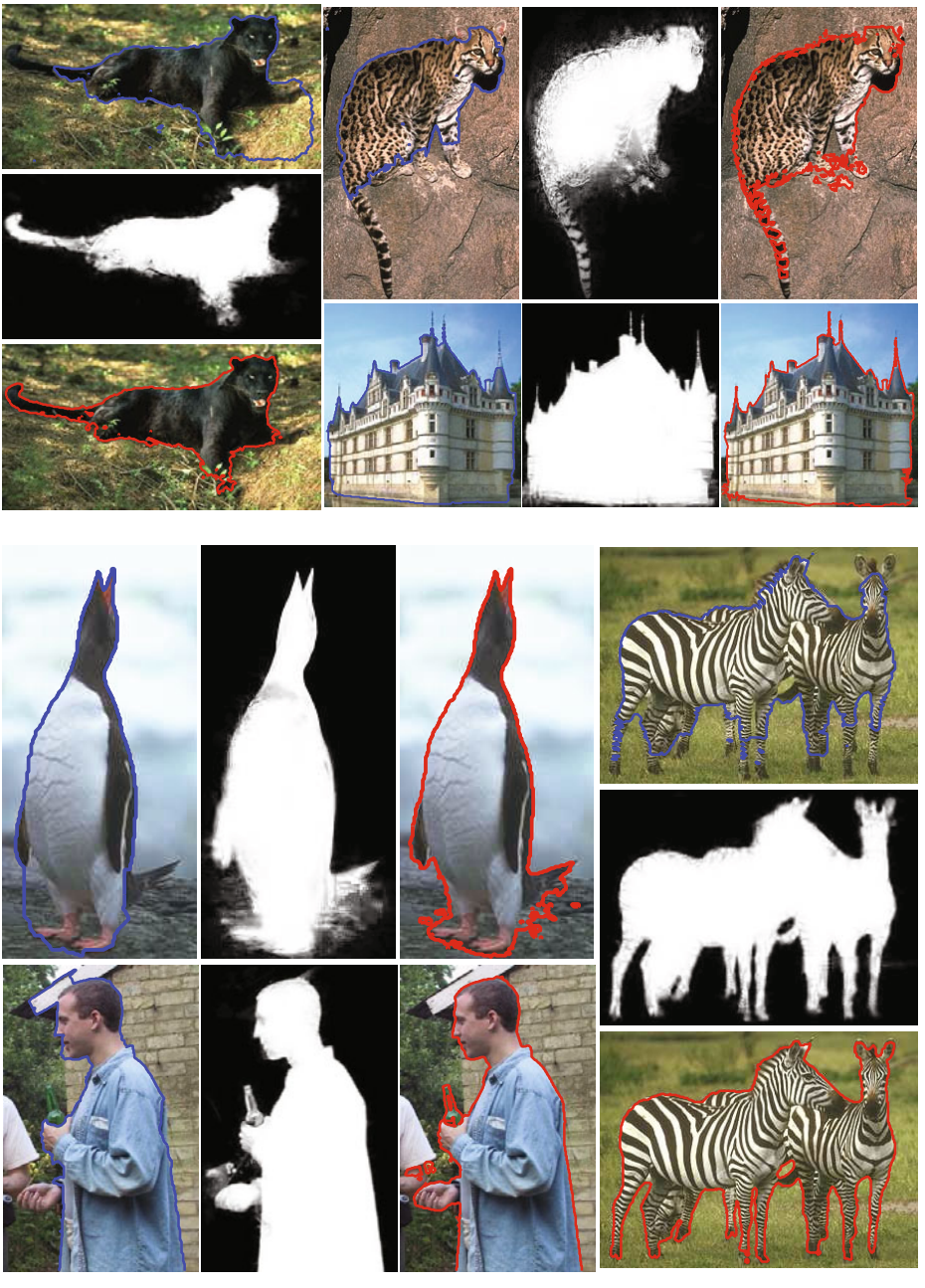


Fig. 4. Segmentation results for GrabCut [21] (*blue contour*) and our new formulation (*red contour*). The grayscale image shows the raw results (the minimum of Eq. (32)) before thresholding. The new formulation shows accurate results for thin parts and blurry boundaries.

upon the normalization of data terms: $\exp(-f_i(0)) + \exp(-f_i(1)) = 1$. For the calculation of L, 5×5 windows are used. Then, we tested two methods on the dataset of [20].

As expected, we could see the performance gain especially in the presence of fuzzy or blurry boundaries. However, the overall error stayed almost same, because the new formulation is sensitive to non-Gaussian noise; the image compression noise often affects the result largely because GrabCut framework relies on iterative estimation. The qualitative comparison confirmed that the new method is very promising. Fig. 4 presents some of the results on the segmentation dataset of [16,20].

6 Conclusion

We presented a novel segmentation framework based on convex optimization by extending graph cuts and closed form matting methods: we obtained a unified viewpoint to see segmentation and matting. While many of the previous works have focused on how to refine the cost function within the fixed forms, e.g. trying various data terms and smoothness terms, the new formulation suggests that we may consider altering the form itself. By doing so, we can overcome the inherent limitation imposed by a certain form.

Encoding new types of prior would be good future research, since some of the recent works showed that incorporating proper prior often boosts the performance: e.g. bounding box prior [14], geodesic star convexity [11], and connectivity prior [25]. It is interesting that many of them have convex objectives.

Hoping it to be useful for solving large scale problems, we release the reference implementation[1] of the primal-dual interior point method that exploits GPU computing technology.

Acknowledgement. The ICT at Seoul National University provides research facilities for this study.

References

1. Argyriou, A., Evgeniou, T., Pontil, M.: Multi-task feature learning. In: Advances in Neural Information Processing Systems, vol. 19 (2007)
2. Bhusnurmath, A., Taylor, C.J.: Graph cuts via l_1 norm minimization. IEEE Trans. PAMI 30, 1866–1871 (2008)
3. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, New York (2004)
4. Buatois, L., Caumon, G., Levy, B.: Concurrent number cruncher: an efficient sparse linear solver on the GPU. In: High Performance Computation Conference (2007)
5. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. PAMI 23, 2001 (1999)
6. Ford, L.R., Fulkerson, D.R.: Maximal flow through a network. Canad. J. Math. 8, 399–404 (1956)

[1] http://ailab.snu.ac.kr/\char‘~yjpark

7. Fu, H., Ng, M.K., Nikolova, M., Barlow, J.L.: Efficient minimization methods of mixed l2-l1 and l1-l1 norms for image restoration. SIAM J. Sci. Comput. 27 (2006)
8. Goldberg, A.V., Tarjan, R.E.: A new approach to the maximum flow problem. In: Eighteenth Annual ACM Symposium on Theory of Computing, pp. 136–146 (1986)
9. Grady, L.: Random walks for image segmentation. IEEE Trans. PAMI 28(11), 1768–1783 (2006)
10. Greig, D.M., Porteous, B.T., Seheult, A.H.: Exact maximum a posteriori estimation for binary images. Journal of the Royal Statistical Society (1989)
11. Gulshan, V., Rother, C., Criminisi, A., Blake, A., Zisserman, A.: Geodesic star convexity for interactive image segmentation. In: CVPR (2010)
12. He, K., Sun, J., Tang, X.: Fast matting using large kernel matting laplacian matrices. In: CVPR (2010)
13. Kim, S., Koh, K., Lustig, M., Boyd, S., Gorinevsky, D.: An interior-point method for large-scale l1-regularized least squares. IEEE Journal of Selected Topics in Signal Processing 1, 606–617 (2007)
14. Lempitsky, V., Kohli, P., Rother, C., Sharp, T.: Image segmentation with a bounding box prior. In: CVPR (2009)
15. Levin, A., Lischinski, D., Weiss, Y.: A closed-form solution to natural image matting. IEEE Trans. PAMI (2008)
16. 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: ICCV (2001)
17. Rhemann, C., Rother, C., Gelautz, M.: Improving color modeling for alpha matting. In: BMVC (2008)
18. Rhemann, C., Rother, C., Rav-Acha, A., Sharp, T.: High resolution matting via interactive trimap segmentation. In: CVPR, pp. 1–8 (2008)
19. Rhemann, C., Rother, C., Wang, J., Gelautz, M., Kohli, P., Rott, P.: A perceptually motivated online benchmark for image matting. In: CVPR, pp. 1826–1833 (2009)
20. Rother, C.: Grabcut dataset, http://tinyurl.com/grabcut
21. Rother, C., Kolmogorov, V., Blake, A.: "GrabCut": interactive foreground extraction using iterated graph cuts. ACM Trans. Graphics 23, 309–314 (2004)
22. Singaraju, D., Rother, C., Rhemann, C.: New appearance models for natural image matting. In: CVPR, pp. 659–666 (2009)
23. Singaraju, D., Vidal, R.: Interactive image matting for multiple layers. In: CVPR (2008)
24. Sinop, A.K., Grady, L.: A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. In: ICCV, pp. 1–8 (2007)
25. Vicente, S., Kolmogorov, V., Rother, C.: Graph cut based image segmentation with connectivity priors. In: ICCV (2008)
26. Wang, J., Cohen, M.F.: Image and video matting: a survey. Foundations and Trends in Computer Graphics and Vision 3, 97–175 (2007)
27. Wang, J., Cohen, M.F.: Optimized color sampling for robust matting. In: CVPR, pp. 1–8 (2007)

Linear Solvability in the Viewing Graph

Alessandro Rudi, Matia Pizzoli, and Fiora Pirri

Department of Computer and System Sciences, Sapienza University of Rome

Abstract. The Viewing Graph [1] represents several views linked by the corresponding fundamental matrices, estimated pairwise. Given a Viewing Graph, the tuples of consistent camera matrices form a family that we call the Solution Set.

This paper provides a theoretical framework that formalizes different properties of the topology, linear solvability and number of solutions of multi-camera systems. We systematically characterize the topology of the Viewing Graph in terms of its solution set by means of the associated algebraic bilinear system. Based on this characterization, we provide conditions about the linearity and the number of solutions and define an inductively constructible set of topologies which admit a unique linear solution. Camera matrices can thus be retrieved efficiently and large viewing graphs can be handled in a recursive fashion. The results apply to problems such as the projective reconstruction from multiple views or the calibration of camera networks.

1 Introduction

In this paper we extend the notion of solvability for a Viewing Graph as given in [1], namely the graph relating several views accounted for by fundamental matrices. We introduce the notion of Solution Set together with a new taxonomy taking into account both linear solvability and the number of solutions of the Viewing Graph. In particular we introduce an inductively constructible set of topologies admitting a unique linear solution. We also show that the method provided allows for a building blocks design that can be used to inductively construct more complex topologies.

Inductive topologies are very useful to combine global and incremental methods for camera matrix estimation. Our formalisation, beside providing a general theoretical framework, it contributes to the hierarchical and recursive approaches for solving the n-view problem in 3D reconstruction and, building on linear-solving subgraphs, it does not involve choosing between multiple solutions or disambiguating results.

Relying on multiview tensors requires establishing feature correspondences between pairs [2], triples [3] or quadruples [4] of images. The geometry of a n-focal system cannot be described by a tensor for $n > 4$. The fundamental matrix is the most basic tool to estimate camera matrices and it can be used as building block to address complex configurations [5]. It can be conveniently estimated from pairs of views and constitutes a less redundant representation

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 369–381, 2011.

than higher order tensors. On the other hand, fundamental matrices for triples of views should be compatible [6], a condition that is satisfied in the case of the trifocal tensor.

Automatic approaches to the computation of structure and motion involve an estimation of camera matrices to initialise the bundle adjustment [7]. The problem of how to initialise the estimation, the order and techniques by which views can be added to the existing ones has been widely faced in literature and several solutions have been proposed. In particular, global methods, based on factorisation [8,9], allow for the computation of sets of views. These methods require to compute feature correspondences across the entire sequence. On the other hand, hierarchical approaches [10] overcome this limitation but introduce the need to merge subsequences in consistent projective frames.

The *Viewing Graph* was introduced by Levi and Werman in [1] to model the bifocal constraints from pairs of views. They studied different topologies and obtained methods to linearly compute the fundamental matrices that can complete the graph. In [11] camera triplets from fundamental matrices constitute the basic subgraphs used to inductively solve triangular topologies using the linearity results from the Viewing Graph.

In this work we start from very well known results on the computation of projective camera matrices and provide a theoretical framework with two aims: to collect and systematise the above results about topologies and linear solvability in a common formalisation and to characterise the topologies for which the estimation is linear and admits a unique solution. Thus, the proposed method is particularly suitable when dealing with the high changeability and severe occlusions that characterise for example the unstructured image datasets collected by web crawling. The paper is organised as follows. We first introduce the *Viewing Graph* and define the *Extended* and *Non Redundant Solution Set*, giving also a sufficient condition for their non-emptiness. In Section 3 a characterisation of the two solution sets is derived in terms of the *Viewing Graph System*; different conditions on the number of solutions and on the linear solvability are formulated and the *Linear Minimal Solution Superset* and the *Linear Maximal Viewing Subgraph* are introduced. Both the taxonomy of the Viewing Graphs and the inductive construction of *Linearly Solvable Viewing Graphs* are addressed in Section 4. Finally, in Section 5, the application to projective reconstruction is sketched and the conclusions are drawn.

2 The Viewing Graph

According to [1] an N-view scene can be represented as a graph $\mathcal{G} = (V, E)$ whose nodes V are the views and whose edges E are the fundamental matrices between the views. Levi and Werman in [1] are concerned with the following problems: 1. Given a subset $E' \subset E$, what further edges can be computed using only E'? 2. Which are the graphs $\mathcal{G}$ such that, given $\mathcal{G}$ and $E' \subset E$, E can be identified univocally? They give algorithms to solve graphs up to 6 views.

2.1 The Solution Set of the Viewing Graph

We introduce here a weaker notion of solving graph [1], namely, the notion of *Solution Set* of a Viewing Graph.

Definition 1 (Solution Set). *Given a viewing graph $\mathcal{G} = (V, E)$, a solution set is the set of n-tuples of camera matrices $\langle P_1, \ldots, P_n \rangle$ which satisfy the constraints associated with the fundamental matrices F_{ij} in E.*

Let $\wp_{\mathcal{G}}$ be the set of all n-tuples of camera matrices which solve the Viewing Graph $\mathcal{G}$. We have that, for any projective transformation Z and for any n-tuple $t \in \wp_{\mathcal{G}}$, the tuple tZ, obtained applying Z to any camera matrix in t, is a solution for the system and so is $tZ \in \wp_{\mathcal{G}}$. Moreover, since any projective matrix is defined modulo a scale factor, if we have that $t = (P_1, ..., P_n) \in \wp_{\mathcal{G}}$ then we have that $t' = (\lambda_1 P_1, ..., \lambda_n P_n) \in \wp_{\mathcal{G}}$ for any $\lambda_i \neq 0$ as well.

In order to avoid these redundancies of representation we are interested in finding only the set $\Psi_{\mathcal{G}} = (\wp_{\mathcal{G}}/GL(4))/\mathbb{R}^{\star}$ of all the orbits of $\wp_{\mathcal{G}}$ under the action of the group of projective transformations and element-wise multiplication by a scalar.

Definition 2 (Extended and Non-Redundant Solution Set). *Given a Viewing Graph $\mathcal{G}$, the* extended Solution Set *is the set of all n-tuples of camera matrices which satisfy the constraints imposed by the fundamental matrices in E. We denote this set by $\wp_{\mathcal{G}}$. The quotient set of $\wp_{\mathcal{G}}$, with respect to projective transformation and scalar multiplication, is the* non-redundant Solution Set*, which we denote by $\Psi_{\mathcal{G}}$.*

Note that, for practical purposes, only $\Psi_{\mathcal{G}}$ is of interest.

We shall state now a sufficient condition for the existence of a solution set. We recall that three fundamental matrices F_{ij}, F_{kj} and F_{ki} are said to be *compatible* if they satisfy the following conditions:

$$\mathbf{e}_{ik}^{\top} F_{ij} \mathbf{e}_{jk} = \mathbf{e}_{kj}^{\top} F_{ki} \mathbf{e}_{ij} = \mathbf{e}_{ki}^{\top} F_{kj} \mathbf{e}_{ji} = 0 \tag{1}$$

with $\mathbf{e}_{ij} \neq \mathbf{e}_{ik}$ the non collinearity conditions for the camera centers [6]. Here $\mathbf{e}_{ij}$ is the epipole arising in view i from view j.

Theorem 1 (Existence of a solution). *Let $\mathcal{G} = (V, E)$ be a Viewing Graph on n views. If all the triples of fundamental matrices satisfy the compatibility condition (1) then there exists a non empty solution set for $\mathcal{G}$.*

Proof. Let $\mathcal{F}_{ijk}$ be the set $\{F_{ij}, F_{ik}, F_{jk}\}$ related to the views v_i v_j and v_k. Note that, from any initial set of compatible triples we can build a solution for $\mathcal{G}$ recursively starting from a triple of fundamental matrices, finding its solution and then adding to the solution an unsolved view at a time. The construction of a solution is illustrated in Section 4.1. □

In the following a Viewing Graph G, in which all triples of fundamental matrices satisfy the compatibility condition, is said to be a *fully compatible* Viewing Graph.

3 The Viewing Graph System

In this section we give a characterisation of the *Extended* and *Non-Redundant* solution sets of a viewing graph and, in particular, we show both the conditions to obtain a linear solution and a simple computation method.

Let us consider Triggs's F-e closure [12]:

$$\textbf{Two views equation:} \qquad F_{12}\, P_1 + \gamma_{12} \left[\mathbf{e}'_{12}\right]_\times P_2 = 0 \tag{2}$$

where F_{12} is the fundamental matrix relating the camera matrices P_1 and P_2, $\mathbf{e}'_{12}$ is the left epipole, $[\cdot]_\times$ is the cross matrix operator and γ_{12} is a free scale parameter. Since the scale parameter is made explicit, the equation carries 8 constraints and 23 degrees of freedom due to the two camera matrices P_1, P_2, including their scale and the overall scale parameter γ_{12}. The *two views equation* (2) completely defines two camera matrices given a fundamental matrix, modulo a projective transformation.

3.1 Characterisation of the Extended and Non-redundant Solution Sets

Let $\mathcal{G} = (V, E)$ be a Viewing Graph and, for homogeneity of representation, let us define $A_{ij} = F_{ij}$ and $B_{ij} = \left[\mathbf{e}'_{ij}\right]_\times$, for any $F_{ij} \in E$. The extended solution set $\wp_\mathcal{G}$ is characterised by the set of two views equations for any F_{ij} related to the constraints it imposes, namely:

$$\wp_\mathcal{G} = \left\{(P_1, \dots, P_n) \mid A_{ij}P_i + \gamma_{ij}B_{ij}P_j = 0, \forall ij \text{ such that there exists } F_{ij} \in E\right\} \tag{3}$$

On the other hand, the quotient set $\Psi_\mathcal{G}$ can be characterised choosing an arbitrary projective frame to which all the solutions should belong. In particular, let $F_{12} \in E$, we select the projective frame in which $P_1 = [I|0]$ and $P_2 = \left[\left[\mathbf{e}'_{12}\right]_\times F_{12} | \mathbf{e}'_{12}\right]$.

Moreover, since any projective matrix is defined modulo a scale factor, for any P_i one of the γ_{ij} is redundant, hence for any i one of γ_{ij} can be set to 1. The non-redundant system characterising the set $\Psi_\mathcal{G}$ is:

$$\begin{cases} A_{ij}P_i + \gamma_{ij}B_{ij}P_j = 0, \forall ij \text{ such that there exists } F_{ij} \in E, \text{ with } F_{ij} \neq F_{12}\} \\ P_2 = \left[\left[\mathbf{e}'_{12}\right]_\times F_{12} | \mathbf{e}'_{12}\right] \\ P_1 = [I|0] \\ \gamma_{i\,k(i)} = 1 \qquad\qquad \forall i \in \{3..n\} \end{cases} \tag{4}$$

here $k(i)$ is a total function defined on $\{3..n\}$.

As we can see, the characterisation of the two Solution sets (3)(4) are algebraic bilinear systems for which, in general, there is no known solution except for very simple special cases [13]

Definition 3 (Linear and Non-Linear Viewing Graph System). *Given a non-redundant Solution Set $\Psi_\mathcal{G}$, the system associated with $\Psi_\mathcal{G}$ is the* Viewing

Graph System, $\Sigma_{\mathcal{G}}$. *Whenever* $\Sigma_{\mathcal{G}}$ *is linearly solvable then* $\Sigma_{\mathcal{G}}$ *and* $\mathcal{G}$ *are, respectively, the* Linear Viewing Graph System *and the* Linear Viewing Graph. *Given a Two Views Equation* $E_{uv} \in \Sigma_{\mathcal{G}}$*, of a Linear Viewing Graph System, this must be equivalent to one of the following three forms:*

$$\begin{array}{ll} \Omega_{ij} & : A_{ij}P_i + B_{ij}P_j \quad = 0 \\ \Delta_{\kappa l} & : A_{\kappa l}P_\kappa + B_{\kappa l}P_l \quad = 0 \\ \Lambda_{i\kappa} & : A_{i\kappa}P_i + \gamma_{i\kappa}B_{i\kappa}P_\kappa = 0 \end{array} \tag{5}$$

Here $u, v \in \{1, \ldots, n\}, k = \{1, 2\}, i, j \in \{3, \ldots, n\}$ *and* P_k *the two constant projective matrices.*

When $\Sigma_{\mathcal{G}}$ *and* $\mathcal{G}$ *are non-linearly solvable they are, respectively, the* Non-linear Viewing System *and the* Non-Linear Viewing Graph.

An example of this kind of systems (and of the choice of $k(i)$) is illustrated in Section 4 for the Base Case I, II and III.

At this point we are ready to discuss the condition for a Viewing Graph System to be linear, the number of solutions and the induced properties on the Viewing Graphs.

Theorem 2 (Unique solution for the Viewing Graph Topology). *Let* $\mathcal{G}$ *be a Viewing Graph of* m *edges and* $n + 2$ *views. Let* $\Sigma_{\mathcal{G}}$ *be the related non-redundant Viewing Graph System.*
If $\Sigma_{\mathcal{G}}$ *has a unique solution then* $m \geq \lceil \frac{11}{7}n - \frac{15}{7} \rceil$.

Proof. $\Sigma_{\mathcal{G}}$ has, by definition, m Two Views Equations in n unknown projective matrices and $m - n$ unknown scale parameters as in (4). Let χ_Σ and δ_Σ be, respectively, the number of constraints and degrees of freedom of $\Sigma_{\mathcal{G}}$. We note that $\chi_\Sigma = 8m$ because any Two View Equation carries 8 constraints and $\delta_\Sigma = 12n + 1(m - n)$, due to the unknown projective matrices and scale factors. Hence $7m \geq 11n$. Note that the inequality states that there should be enough fundamental matrices in order to constrain the degrees of freedom of the unknown camera matrices. Therefore if $\Sigma_{\mathcal{G}}$ has a unique solution it must be also that $\mathcal{G}$ has $m \geq \lceil \frac{11}{7}n - \frac{15}{7} \rceil$ edges. □

We show, now, the conditions for a viewing graph to be linearly solvable. Consider Figure 1, this illustrates a Viewing Graph that can undergo a construction

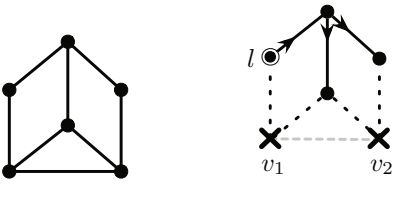


Fig. 1. The Graph $\mathcal{G}$, on the left, satisfies the condition of linear solvability. On the right the construction required by the condition.

inducing linear solvability. Linear solvability, indeed, requires a suitable assignment of the scale factor, therefore the Viewing Graph that can be adjusted so as to guarantee this assignment is linearly solvable. More specifically, consider the two constant views v_1 and v_2 associated with the constant cameras, which therefore must be connected by an edge, and their neighbours. Let us define the *open neighbour set* $n(v_1, v_2)$ to be the set of all nodes connected to v_1 and v_2, excluding v_1 and v_2. Let us cut all the edges between any node $v_h \in n(v_1, v_2)$ and v_1, v_2. Take any node $l \in n(v_1, v_2)$. If it is possible, starting from l to build a unique path from l to all the nodes in $V - \{v_1, v_2\}$ then we can use this path to ensure that the scale factor is inherited, through l, to all the nodes. If this construction is possible, this means that the graph $\mathcal{H}(v_1, v_2)$, illustrated on the right of Figure 1 can be obtained, hence the starting graph $\mathcal{G}$, as illustrated on the left of the figure, is a linear Viewing Graph.

More formally:

Theorem 3. *Let $\mathcal{G} = (V, E)$ be a Viewing Graph with $n \geq 3$ views. Let $n(v_1, v_2)$ be the union of the open neighbourhood of $v_1, v_2 \in V$, let $\mathcal{H}(v_1, v_2) = \mathcal{G}[V - \{v_1, v_2\}]$ the induced subgraph of $\mathcal{G}$ over the vertex set $V - \{v_1, v_2\}$.*

If there exist $v_1, v_2 \in V$, $l \in n(v_1, v_2)$ and a unique orientation for each edge in $\mathcal{H}(v_1, v_2)$ such that:

- *the in-degree of any node in $V - \{v_1, v_2, l\}$ is 1 and the in-degree of l is 0,*

then $\mathcal{G}$ is a Linearly Solvable Viewing Graph.

Proof. We show the statement by constructing directly the linear system according to Definition 3 Now we prove by construction that if exist $v_1, v_2, l \in V$ and $\mathcal{H}(v_1, v_2)$ that satisfy the hypothesis, then the Viewing Graph System associated to $\mathcal{G}$ is linear.

Suppose that $v_1, v_2, l \in V$ and $\mathcal{H}(v_1, v_2)$ exist and satisfy the statement. First of all, we note that $\mathcal{H}(v_1, v_2)$ has at most $n - 3$ edges, because any node except l can have at most an entering edge and the nodes in $\mathcal{H}(v_1, v_2)$ are $n - 2$. Then we proceed in the construction of a Viewing Graph System Σ. We start from a system Σ which contains only the equations $P_2 = \left[[e'_{12}]_\times F_{12} | e'_{12}\right]$ and $P_1 = [I|0]$ For any edge e_{ij} (edge from v_i to v_j) in $\mathcal{H}(v_1, v_2)$ we add in Σ the equation Ω_{ij}. We choose an edge from $\mathcal{G}$ which connects l to v_κ, with $\kappa \in \{1, 2\}$ and add to Σ the equation $\Delta_{\kappa l}$. For any other edge $e_{i\kappa}$ which connects the nodes v_1, v_2 to their neighbourhood $n(v_1, v_2)$ we add to Σ the equation $\Lambda_{i\kappa}$. The equations Ω, Δ, Λ are, thus, as in Definition 3. The system, specified by Ω, Δ, Λ, mentions the two equations defining P_1 and P_2. For any edge in $\mathcal{G}$ there is the related equation in $\Sigma_\mathcal{G}$ and for any camera matrix P_i, with $i = 3 \ldots n$, the scale parameter is set to 1 (the missing parameter γ_{ij} in the equation Ω_{ij} and $\gamma_{\kappa l}$ in the equation $\Delta_{\kappa l}$).

Thus, according to Definition 3, $\Sigma_\mathcal{G}$ is a linear Viewing Graph System. □

We can note that this sufficient condition allows us to speak directly about linear solvability of a Viewing Graph $\mathcal{G}$ from a topological point of view without using the related algebraic representation $\Sigma_\mathcal{G}$.

Resolution of a general linear Viewing Graph System. Let Σ be a non-redundant Viewing Graph System composed by m Two Views Equations in n unknown camera matrices $P_3, \ldots, P_{n+3}$ and $r = n - m$ scale parameters represented by the vector $\Gamma = \left(\gamma_{g(1)}, \ldots, \gamma_{g(r)}\right)$, where g is a function which associates the position in the vector Γ to the indices of the fundamental matrix related to that scale parameter and g^{-1} its inverse. The solutions can be found vectorising the system such that it assumes the form $Ax = b$ where A and b are suitable constant matrices and x is the unknown vector. In that case the space of all solutions is $x = A^{+}b + null(A)z$, with z a free vector of suitable dimension.

In order to transform Σ in the form $Ax = b$ first of all we represent the unknowns of Σ in the form of a vector $x = \left(vec(P_3)^{\top}, \ldots, vec(P_{n+3})^{\top}, \gamma_{g(1)}, \ldots, \gamma_{g(r)}\right)^{\top}$ of length $12n + r$ and define two matrices $selP_i = (\varphi_{n,i} \otimes I_{12\times 12}, \mathbf{0}_r)$, $sel\gamma_{ij} = \left(\mathbf{0}_{12n}, \varphi_{r,g^{-1}(ij)}\right)$ useful to select respectively only the matrix P_i from x and only the scale parameter γ_{ij}. Here vec is the vectorisation operator, $\otimes$ is the Kronecker product, $\varphi_{n,i}$ is a row vector of length n with all components equal to 0 excepts the i-th which is 1, $\mathbf{0}_n$ is the row vectors of length n of all zeros and $I_{n\times n}$ the identity matrix of dimension $n \times n$.

Thus, to transform Σ in the form $Ax = b$, we simply rewrite the equation in (5) of Definition 3, with respect to x as follows:

$$\begin{aligned} \Omega_{ij} &\equiv \{(I_{4\times 4} \otimes A_{ij})\, selP_i \;+\; (I_{4\times 4} \otimes B_{ij})\, selP_j\}\, x = 0 \\ \Delta_{\kappa i} &\equiv \{(I_{4\times 4} \otimes B_{\kappa i})\, selP_i\}\; x = vec\,(A_{\kappa i} P_\kappa) \\ \Lambda_{i\kappa} &\equiv \{(I_{4\times 4} \otimes A_{i\kappa})\, selP_i \;+\; vec\,(B_{i\kappa} P_\kappa)\, sel\gamma_{i\kappa}\}\, x = 0 \end{aligned} \tag{6}$$

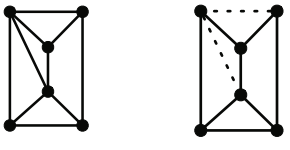


Fig. 2. Given a nonlinear Viewing Graph $\mathcal{G}$ on the left, the associated *Linear Maximal Viewing Subgraph* $\Gamma^{\star}_{\mathcal{G}}$ on the right.

3.2 The Linear Minimal Solution Superset of a Nonlinear Viewing Graph

Given a Nonlinear Viewing Graph $\mathcal{G}$ we are interested in simplifying as much as possible the search for the solution set $\Psi_{\mathcal{G}}$. In order to do so we introduce the Linear Minimal Superset of $\Psi_{\mathcal{G}}$ and the Linear Maximal Viewing Subgraph of $\mathcal{G}$.

Given a Viewing Graph $\mathcal{G} = (V, E)$ and the associated Viewing Graph System $\Sigma_{\mathcal{G}}$, an edge e in $\mathcal{G}$ corresponds to a constraint c in $\Sigma_{\mathcal{G}}$. This means that removing e from $\mathcal{G}$, and so obtaining $\mathcal{G}' = (V, E - \{e\})$, then the corresponding Viewing Graph System $\Sigma_{\mathcal{G}'}$ is less constrained than $\Sigma_{\mathcal{G}}$ indeed $\Sigma_{\mathcal{G}'} = \Sigma_{\mathcal{G}} - \{c\}$. Thus $\Psi_{\mathcal{G}} \subseteq \Psi_{\mathcal{G}'}$.

Definition 4 (Linear Maximal Viewing Subgraph). *Given a Viewing Graph $\mathcal{G}$, and letting $lin(\mathcal{G})$ be the set of all the Linear Viewing Subgraphs of $\mathcal{G}$, the Linear Maximal Viewing Subgraph of $\mathcal{G}$ is the Viewing Graph $\Gamma^\star_\mathcal{G} \in lin(\mathcal{G})$, such that the associated linear solution set is minimal :*

$$\Gamma^\star_\mathcal{G} = \arg \min_{\Gamma \in lin(\mathcal{G})} |\Psi_\Gamma|. \tag{7}$$

See Fig. 2. The definition is well-posed, indeed given a Nonlinear Viewing Graph $\mathcal{G} = (V, E)$ we can always remove edges from $\mathcal{G}$ in order to obtain a linearly solvable Viewing Subgraph $\mathcal{L} = (V, E')$ with $E' \subset E$ and so $\Psi_\mathcal{G} \subseteq \Psi_\mathcal{L}$. This means that a Linear Viewing Subgraph of a Viewing Graph System $\mathcal{G}$ always exists (for example $\mathcal{L} = (V, \emptyset)$).

Definition 5 (Linear Minimal Solution Superset). *Given a Viewing Graph $\mathcal{G}$, the Linear Minimal Solution Superset of $\mathcal{G}$, $\Pi_\mathcal{G}$, is the Solution Set of the Linear Maximal Viewing Subgraph $\Gamma^\star_\mathcal{G}$*

$$\Pi_\mathcal{G} = \Psi_{\Gamma^\star_\mathcal{G}} \tag{8}$$

The Linear Minimal Solution Superset of a nonlinear Viewing Graph is important because we have always that $\Psi_\mathcal{G} \subseteq \Pi_\mathcal{G}$ and $\Pi_\mathcal{G}$ is linearly computable. As a consequence, when we are searching for a solution to $\mathcal{G}$ we have only to search in the minimal linear space $\Pi_\mathcal{G}$ instead of in the huge Cartesian product of all camera matrices.

Moreover, when $\Pi_\mathcal{G}$ contains only a solution and the fundamental matrices expressed by $\mathcal{G}$ are fully compatible, we have that $|\Pi_\mathcal{G}| = 1$, $|\Psi_\mathcal{G}| \geq 1$, $\Psi_\mathcal{G} \subseteq \Pi_\mathcal{G}$ and thus $\Psi_\mathcal{G} = \Pi_\mathcal{G}$. This means that in this case we have only to solve the linear graph $\Gamma^\star_\mathcal{G}$ to have the solution of the bigger and possibly nonlinear $\mathcal{G}$.

4 Topology and Solvability

First of all we collect from the previous sections some results about the Topology of a Viewing Graph, the linearity and the number of solutions of the related Viewing Graph System.

Any Viewing Graph with m edges and n nodes where $m < \left\lceil \frac{11}{7}n - \frac{15}{7} \right\rceil$ and any linear Viewing Graph whose system is underdetermined has a family of solutions in the Solution Set.

Any full compatible linear or nonlinear Viewing Graph $\mathcal{G}$ for which $|\Pi_\mathcal{G}| = 1$ admits only one linear solution.

Finally we can introduce the taxonomy of the set of Viewing Graphs in terms of number of solutions and linear solvability (fig. 3).

4.1 An Inductively Constructible Set of One Solution Linear Viewing Graphs

If we are able to solve a linear Viewing Graph Γ which admits a unique solution then we are able to solve all Viewing Graphs for which Γ is the Linear Maximal Viewing Subgraph.

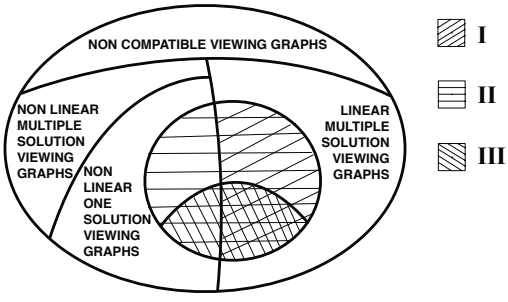


Fig. 3. Taxonomy of the Set of Viewing Graphs. I) One Solution Linear Viewing Graphs; II) Viewing Graphs whose Linear Maximal Subgraph has one solution; III) Viewing Graphs whose Linear Maximal Subgraph belongs to the set of the Recursive Topologies we presented in subsection 4.1.

Thus, it is important to find a cheap way to span the space of topologies that lead to Linear Viewing Graphs with unique solution. With this aim, in the following we give a characterization of a constructible subset of the set of the *One Solution Linear Viewing Graphs*.

We apply the results from the previous sections.

Base Linear Case I: Three Views. Consider the Viewing Graph $\mathcal{G} = (V, E)$ given by three views $V = (v_1, v_2, v_3)$ and the three fundamental matrices linking them $E = (F_{12}, F_{23}, F_{31})$ (fig. 4). This topology satisfies the necessary and sufficient conditions in section 3 and so it is linear. In addition, when the camera matrices are in general position, it admits at most one solution. When it satisfies the full compatibility condition then $\mathcal{G}$ has a unique solution. This is a well known result [6,1]. The interesting thing here is that we demonstrated it only using the tools from the previous sections.

For completeness, we algebraically find a close solution for $\mathcal{G}$. The orbit set of solutions $\Psi_{\mathcal{G}}$ is represented by the related non-redundant Viewing Graph System as follows:

$$\begin{cases} A_{23}P_2 + B_{23}P_3 = 0 \\ A_{31}P_3 + \gamma_{31}B_{13}P_1 = 0 \\ P_1 = [I|0] \\ P_2 = \left[[\mathbf{e}'_{12}]_\times F_{12}|\mathbf{e}'_{12}\right] \end{cases} \tag{9}$$

where in the first equation γ_{23} has been set to 1 in order to disambiguate the scale of P_3 with respect of P_2 (see proof of the sufficiency condition in Section 2.1). Thus the system is linear; in general it is overdetermined but, when the fundamental matrices are compatible, it admits at least one solution which can be stated in closed form:

$$\begin{cases} P_1 = [I|0] \\ P_2 = \left[[e'_{12}]_\times F_{12}|e'_{12}\right] \\ P_3 = \begin{pmatrix} B_{23} \\ A_{31} \end{pmatrix}^+ \begin{pmatrix} A_{23}P_2 \\ \gamma_{23}B_{31}P_1 \end{pmatrix} \end{cases} \tag{10}$$

where $(\cdot)^+$ is the pseudo-inverse operator and $\gamma_{23} = -\frac{cd^\top}{dd^\top}$, with $c = UA_{23}P_2$, $d = VB_{31}P_1$ and $(U\,V) = null\begin{pmatrix} B_{23} \\ A_{31} \end{pmatrix}$.

Base Linear Case II: Five Views. Let $\mathcal{G} = (V, E)$ be the Viewing Graph with $V = (v_1, \ldots, v_5)$ and $E = (F_{12}, F_{31}, F_{41}, F_{25}, F_{35}, F_{45})$ as in figure (fig. 4). This topology satisfy the conditions of section 3 so that it's linear and, when the camera matrices are in general position, it admits at most one solution. When it satisfies the full compatibility condition, then $\mathcal{G}$ has a unique solution [1].

For completeness: the set of solutions $\Psi_\mathcal{G}$ is represented by the following Linear Viewing Graph System

$$\begin{cases} A_{31}P_3 + \gamma_{31}B_{31}P_1 = 0 \\ A_{41}P_4 + \gamma_{41}B_{41}P_1 = 0 \\ A_{25}P_2 + B_{25}P_5 = 0 \\ A_{54}P_5 + B_{54}P_4 = 0 \\ A_{53}P_5 + B_{53}P_3 = 0 \\ P_1 = [I|0] \\ P_2 = \left[[e'_{12}]_\times F_{12}|e'_{12}\right] \end{cases} \tag{11}$$

here $\gamma_{25}, \gamma_{54}, \gamma_{53}$ have been set to 1 in order to disambiguate the scale respectively of P_5 with respect to P_2, P_4 with respect to P_5 and P_3 with respect to P_5 (see the proof in Section 2.1).

Base Linear Case III: Six Views. Let $\mathcal{G} = (V, E)$ be the Viewing Graph with $V = (v_1, \ldots, v_6)$ and $E = (F_{32}, F_{52}, F_{61}, F_{12}, F_{13}, F_{34}, F_{45}, F_{56})$ as in figure (fig. 4).

Even in this case $\mathcal{G}$ satisfies the two topological conditions of section 2.1. Consequently, it is linearly solvable. If the camera matrices are in general positions it admits at most a solution which exist when the full compatibility holds (see [1])

For completeness: the set of solutions $\Psi_\mathcal{G}$ is represented by the following Linear Viewing Graph System

$$\begin{cases} A_{32}P_3 + \gamma_{32}B_{32}P_2 = 0 \\ A_{52}P_5 + \gamma_{52}B_{52}P_2 = 0 \\ A_{61}P_6 + \gamma_{61}B_{61}P_1 = 0 \\ A_{13}P_1 + B_{13}P_3 = 0 \\ A_{34}P_3 + B_{34}P_4 = 0 \\ A_{45}P_4 + B_{45}P_5 = 0 \\ A_{56}P_5 + B_{56}P_6 = 0 \\ P_1 = [I|0] \\ P_2 = \left[[\mathbf{e}'_{12}]_\times F_{12}|\mathbf{e}'_{12}\right] \end{cases} \tag{12}$$

here $\gamma_{13}, \gamma_{34}, \gamma_{45}, \gamma_{56}$ have been set to 1 in order to disambiguate the scale respectively of P_3 with respect to P_1, P_4 with respect to P_3, P_5 with respect to P_4 and P_6 with respect to P_5 (see the proof in Section 2.1).

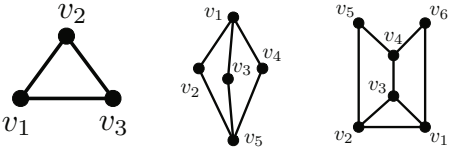


Fig. 4. In order: Base Linear Case I, Case II, Case III

Composition rule	**I**	**II**	**III**	**IV**
	Merging two solved graphs which share two views	Merging two solved graphs which share a view and are linked by an edge	Adding a new double connected view	Merging two solved graphs with three edges

Composition Rule I: Merging two solved graphs which share two views. Let $\mathcal{G}$ be a Viewing Graph composed by two already solved graphs Γ and Υ which share two views v_1 and v_2. The solution of $\mathcal{G}$ $t_{\mathcal{G}}$ can be built from the solutions $t_\Gamma = (P_1, P_2, \dots)$ and $t_\Upsilon = (P'_1, P'_2, \dots)$ (Fig. 5). Indeed, since the cameras P_1, P_2 and P'_1, P'_2 are linked by the same fundamental matrix F_{12}, a projective transformation exists that maps P_i on P'_i for $i = 1, 2$. We can find it simply by

$$Z = \begin{pmatrix} P_1 \\ P_2 \end{pmatrix}^+ \begin{pmatrix} P'_1 \\ P'_2 \end{pmatrix}$$

Once we have Z we have to right multiply all the camera matrices in t_Γ in order to express them in the same projective frame of Υ such that P_i and P'_i coincide for $i = 1, 2$.

Composition Rule II: Merging two solved graphs which share a view and are linked by an edge. Let $\mathcal{G}$ be a Viewing Graph composed by two already solved graphs $\Gamma = (V, E)$ and $\Upsilon = (W, L)$ which share a view v_1 and are linked by a fundamental matrix F_{23} with the related views $v_2 \in V$ and $v_3 \in W$ (fig. 5). We can solve $\mathcal{G}$ by finding the solution to $\Theta = (\{v_1, v_2, v_3\}, \{F_{12}, F_{13}, F_{23}\})$ through Base Case I and then apply Composition Rule I two times, first to Γ, Θ and then to $(\Gamma \cup \Theta), \Upsilon$.

Composition Rule III: Adding a new double connected view. Let $\mathcal{G}$ be a Viewing Graph with n views composed by an already solved graph $\Gamma = (V, E)$ of $n-1$ views connected by F_{1n}, F_{2n} to a view v_n and let $t = (P_1, \ldots, P_{n-1})$ the solution for Γ (Fig. 5). We are able to solve the Viewing Graph $\mathcal{G}$ as follows. First of all we calculate F_{12} by P_1 and P_2 from t. Then we solve the Viewing Graph System related to the Viewing Graph $\Upsilon = (\{v_1, v_2, v_n\}, \{F_{12}, F_{1n}, F_{2n}\})$ using the Base Case I and then we merge the two already solved Viewing Graphs Γ, Υ using the Compositional Rule I.

Composition Rule IV: Merging two solved graphs with three edges. Let $\mathcal{G}$ be a Viewing Graph composed by two already solved graphs $\Gamma = (V, E)$ and $\Upsilon = (W, L)$ which are linked by three fundamental matrices F_{14}, F_{25}, F_{36} with the related views $v_1, v_2, v_3 \in V$ and $v_4, v_5, v_6 \in W$ (Fig. 5). Supposing that Γ and Υ are already solved, we can calculate F_{12}, F_{23}, F_{13} from Γ and F_{45}, F_{56} from Υ. Then we can solve $\mathcal{G}$ by finding the solution to $\Theta = (H, K)$ with $H = \{v_1, v_2, v_3, v_4, v_5, v_6\}$, $K = \{F_{12}, F_{23}, F_{13}, F_{45}, F_{56}, F_{14}, F_{25}, F_{36}\}$ through Base Case III and then apply Composition Rule I two times, first to Γ, Θ and then to $(\Gamma \cup \Theta), \Upsilon$.

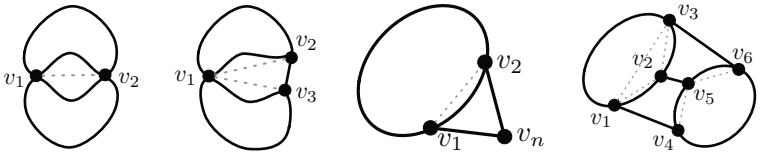


Fig. 5. In order: Compositional Rule I, Rule II, Rule III, Rule IV

In this subsection we substantially sketched a compositional topology that allows to compute the solution to the Linear Maximal Viewing Subgraph of a given Graph $\mathcal{G}$ in a bottom-up fashion. We can solve arbitrarily chosen parts of this subgraph and merge them in arbitrary order (in agreement with the conditions of the composition rules) arriving to the same result, because of the linearity of the problem under the sufficiency condition of Theorem 1, at least in the unique solution case.

In the end, due to this invariance to the order, this incremental bottom-up linear approach makes the problem of choosing a right view order less critical.

5 Conclusions and Discussion

This paper integrates in a common framework previous results on the estimation of camera matrices from unstructured collections of views with a characterization of a subclass of topologies for which the solution is linear and unique. The Viewing Graph has been equipped with its algebraic counterpart, the Viewing Graph System. This characterization provides a sufficient condition for the linear

solvability of the system of bilinear equations and the associated Viewing Graph. Indeed, we translated the linear solvability condition to be directly applied to the topology of Viewing Graphs, bridging from the algebraic to the graph based representation. In future works we are going to use the tools described in this paper to develop heuristic and approximated graph algorithms operating on Viewing Graphs in order to compute the Linear Maximal viewing Subgraph and find the Minimal Solution Set. A similar approach is particularly suitable to deal with the high changeability and severe occlusions that characterize, for example, the large, unstructured image datasets collected by web crawling.

Acknowledgement. This research has been funded by EU-FP7 *NIFTI* project.

References

1. Levi, N., Werman, M.: The viewing graph. In: CVPR (2003)
2. Luong, Q.T., Faugeras, O.: The fundamental matrix: theory, algorithms, and stability analysis. International Journal of Computer Vision 17, 43–75 (1995)
3. Torr, P., Zisserman, A.: Robust parameterization and computation of the trifocal tensor. Image and Vision Computing 15, 591–605 (1997)
4. Hartley, R.I.: Cheirality. Int. J. Comput. Vision 26, 41–61 (1998)
5. Goldberger, J.: Reconstructing camera projection matrices from multiple pairwise overlapping views. Computer Vision and Image Understanding 97, 283–296 (2005)
6. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004) ISBN: 0521540518
7. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment - a modern synthesis. In: ICCV (2000)
8. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. Int. J. Comput. Vision 9, 137–154 (1992)
9. Sturm, P.F., Triggs, B.: A factorization based algorithm for multi-image projective structure and motion. In: Buxton, B.F., Cipolla, R. (eds.) ECCV 1996. LNCS, vol. 1065. Springer, Heidelberg (1996)
10. Fitzgibbon, A.W., Zisserman, A.: Automatic camera recovery for closed or open image sequences. In: Burkhardt, H.-J., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1406, p. 311. Springer, Heidelberg (1998)
11. Sinha, S.N., Pollefeys, M., McMillan, L.: Camera network calibration from dynamic silhouettes. In: CVPR (2004)
12. Triggs, B.: Linear projective reconstruction from matching tensors. Image and Vision Computing 15, 617–625 (1997)
13. Cohen, S., Tomasi, C.: Systems of bilinear equations (1998)

Inference Scene Labeling by Incorporating Object Detection with Explicit Shape Model

Quan Zhou and Wenyu Liu

Dept. of Electronics and Information Engineering,
Huazhong University of Science and Technology, Wuhan, China PR
qzhou.lhi@gmail.com, liuwy@hust.edu.cn

Abstract. In this paper, we incorporate shape detection into contextual scene labeling and make use of both shape, texture, and context information in a graphical representation. We propose a candidacy graph, whose vertices are two types of recognition candidates for either a superpixel or a window patch. The superpixel candidates are generated by a discriminative classifier with textural features as well as the window proposals by a learned deformable templates model in the bottom-up steps. The contextual and competitive interactions between graph vertices, in form of probabilistic connecting edges, are defined by two types of contextual metrics and the overlapping of their image domain, respectively. With this representation, a composite clustering sampling algorithm is proposed to fast search the optimal convergence globally using the Markov Chain Monte Carlo (MCMC). Our approach is applied on both lotus hill institute (LHI) and MSRC public datasets and achieves the state-of-art results.

1 Introduction

As Fig. 1 illustrates, this paper presents an semantic scene understanding (labeling) method, motivated by partitioning or segmenting an entire image in (a) into distinct recognizable regions in (b). This task requires classify all pixels, while preserving accurate segmentation. By generating recognition candidates by superpixel classification and object detection in the bottom-up steps as shown in (c) and (d) respectively, we present a candidacy graphical representation to integrate shape, texture, and context information.

We start by reviewing the literature on two research streams: semantic image segmentation and structural object detection that are related to the bottom-up steps of our work.

(i) Many approaches of image segmentation often explore textural appearance features, and use flat graphical representation to encode local confidence and pairwise consistency. Examples include the methods based on Markov random fields (MRFs) and the conditional random fields (CRFs). The former models the joint probability of the image and its corresponding semantic labels [1], and latter models the conditional probability of the labels [2,3]. Recently, the global and contextual information based on the graphical representation are

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 382–395, 2011.

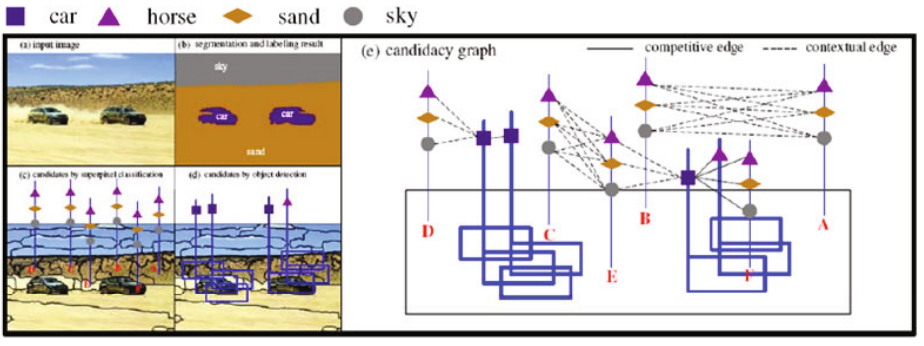


Fig. 1. Illustration of the proposed method. Given an input images in (a), the recognition candidates of over-segmented superpixels (denoted by red letters), as well as the candidates of template-based detector (denoted by blue rectangles), are extracted in (c) and (d) respectively. A candidacy graphical representation is constructed with these candidates as illustrated in (e). The final labeling result is exhibited in (b).

explored by some innovative work [4,5,6,7,8]. The inference task on graphical models can be formulated as energy minimization problems with soft and hard constraints. The algorithms can be divided into deterministic approximation algorithms, such as the graph cuts [9] and the belief propagation (BP) [10], and stochastic algorithms, like constraint-satisfaction solvers [11], Gibbs sampler [12].

(ii) Some other methods are aimed at detect and localize object-of-interest from cluttered scene by capturing shape information. These methods usually represent structural objects by a spatial or context configuration with a small number of primitives, such as the PAS-based model [13], the shape context [14], and the recent proposed active basis model [15].

Though the research in these two streams has made remarkable progress, it still remains a challenge to combine the two types of method for the entire scene understanding due to the difficulty of integrating shape and texture model. A few pioneer work demonstrates this path with some special cases [17,18].

In this paper, we study (i) a candidacy graphical representation that incorporates the shape information and textural appearance in a Bayesian framework (ii) a composite cluster sampling algorithm for energy convergence globally.

Given an input image, we first generate a batch of recognition candidates (proposals) by two types of classifiers: superpixel classifier and shape detector. The superpixel classifier is learned by JointBoost method with a bank of low-level features, inspired by [3], and it gives the possible labels to each superpixel. The active basis model [15] is employed as shape detector to learn deformable templates of structural object, and used to generate possible matchings on the testing image, as shown in Fig. 1 (d). We thus build up an adjacency candidacy graphical representation with these candidates, as illustrated in Fig. 1 (e) and Fig. 3, where each graph vertex is equivalent to a recognition candidate. Each two vertices can be linked by a probabilistic edge denoting the competitive or

contextual interaction. Thus the semantic parsing can be solved by validating these candidates while accounting for the interactions among them.

With this representation, we present a composite cluster sampling algorithm using the Markov Chain Monte Carlo (MCMC) mechanism [19]. Unlike the traditional single-site sampler [11,12], this algorithm updates large portions of the solution space quickly to minimize constraint energy, by clustering connected components in each sampling step. It can be viewed as an extension of the multiple-site sampler [20] by dealing with the soft (contextual) and hard (competitive) constraints simultaneously. Given the candidacy graphical representation, this algorithm contains two iterative steps: (i) Sampling the competitive and contextual edges to form a composite cluster; (ii) Validating the graph vertices of this cluster following the Markov Chain Monte Carlo (MCMC) mechanism [19].

The remainder of this paper is arranged as follows. We first present the bottom-up proposal and candidacy representation in Sect. 2, and follow with a description of the problem formulation in Sect. 3. The inference algorithm is discussed in Sect. 4. The experimental results are shown in Sect. 5 and the paper concludes with a summary in Sect. 6 .

2 Representation

In this section, we first introduce the recognition candidate generation by two types of classifiers and then discuss a candidacy graphical by these candidates.

2.1 Bottom-Up Candidates Generation

Given an input image I, we first use two types of classifier to generate recognition candidates: one for superpixels with low-level (textural appearance) features and the other for structural objects with shape templates. A candidate is defined as a universal form $c_i = (A_i, l_i)$. $A_i = (X_i, \Gamma_i, \Lambda_i, \omega_i)$ denotes the candidate attributes, including location X_i, outer contour Γ_i and image domain Λ_i, respectively. $\omega_i \in \{0, 1\}$ is the binary variable that denotes the validation or not of c_i. $l_i = (' car', ' grass', ' cow', \ldots)$ denotes the semantic label.

We start by discussing generation recognition candidates by these two classifiers.

(i) Superpixel-based candidate.

Superpixels are often used to effectively reduce the solution complexity in image segmentation. In this work, we use an over-segmentation scheme used in [24] to obtain the superpixels of both training and testing images. In practice, each image contains around $30 \sim 50$ superpixels.

Given the annotated training images, we collect a pool of textural features, including texton filters [3], color [22], and location [4], and then learn a discriminative classifier formed by a set of selected features using a boosting framework [23]. In the testing stage, each superpixel receives a recognition score for each semantic label by this classifier, and a batch of superpixel candidates are generated.

Thus the energy cost for each superpixel-based proposal can be computed by

$$E_T(c_i|I) = \sum_{j=1}^{n(T)} \alpha_j f_j(A_i, l_i), \tag{1}$$

where $f_j(A_i, l_i)$ is one selected feature (weak classifier) over superpixel and semantic label, and α_j is the weight parameter. $n(T)$ denotes the number of shape templates.

(ii) Template-based candidate.

A recent proposed active basis model [15] is utilized to capture shape information of structural object categories (car, cow, and horse, etc.). Using the model with a shared sketch algorithm, deformable templates can be learned for each object category on a small set of aligned positive samples in the same pose without negative samples. A template $\mathbf{B}_{l_i}$ for category l_i, consist of a set of active Gabor basis $\{B_j\}$ that are allowed to slightly perturb their locations and orientations before they are linearly combined to generate the image. One can use other object detection approaches [16] without major algorithmic changes.

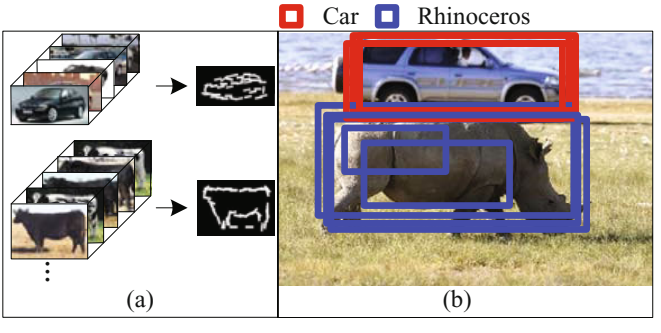


Fig. 2. Template-based recognition candidates generated by active basis model [15]. (a) illustrates the deformable template learning from a set of aligned positive samples and (b) shows an example of template detection.

In Fig. 2 (a), we intuitively illustrate the template learning process from several aligned training images, and Fig. 2 (b) shows an example of detecting the structural object instances by matching the templates in cluttered image, (red rectangle for car detection and blue rectangle for rhinoceros).

We solve the energy cost for each candidate by template matching [15], as

$$E_S(c_i|I) = \sum_{j=1}^{n(S)} [\lambda_j h(| < I(\Lambda_i), B_j > |^2) - \log Z(\lambda_j)], \tag{2}$$

where $h(\cdot)$ is transformation function on filter response. λ_j denotes the learned weight parameter for each basis B_j, and $Z(\lambda_j)$ is the normalizing constant and it can be computed using [15]. $n(S)$ denotes the number of superpixels.

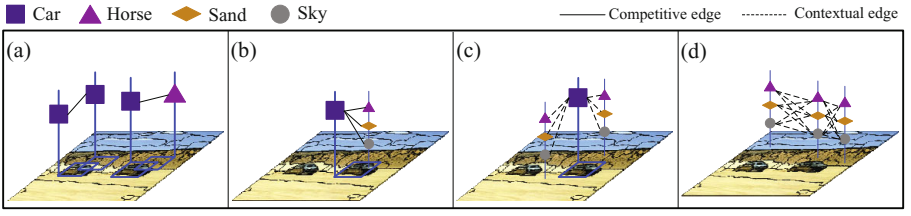


Fig. 3. Incorporating contextual and competitive interactions in the candidacy graphical representation. The blue rectangles indicate the detected object templates. The polygons on thick line denote the template-based candidates and polygons on thin line denote the superpixel-based candidates. The different polygons with different colors denote recognition label. The dashed line and solid line denote the contextual and competitive edge link respectively. The competitive edges exist between two candidates sharing image domain (as shown in (a) and (b)), and the contextual edges for connecting candidates defined by context features (as shown in (c) and (d)).

2.2 Candidacy Graph Construction

With these generated recognition candidates, we establish an adjacency graphical representation $G = (V, E)$, whose vertex v_i is equivalent to a candidate c_i. We thus define the graph vertex set as,

$$V = V_T \bigcup V_S = \{v_i = c_i = (A_i, l_i), i = 1, \ldots n(T) + n(S)\}, \tag{3}$$

where V_T and V_S are candidate sets from superpixel classifier and shape detector, respectively. In this graph, the parsing problem can be formulated as the candidate validating task.

For any two vertices v_i and v_j specified by two adjacent superpixels, a probabilistic edge $e =< v_i, v_j >, e \in E$ is defined to indicate the competitive or contextual interaction between them, this leads to $E = E^+ \cup E^-$. Each **contextual edge** E^+ exists between two vertices that share the contextual correlation, while each **competitive edge** E^- accounts for the mutual exclusion between two vertices that share overlapping in image domain. Fig. 3 illustrates a typical example of the candidacy graphical representation.

Competitive edges are defined for the mutual exclusion constraint that the two vertices should not both be validated if they overlap with each other in image domain. The overlapping often occurs with two neighboring template candidates or one template candidate including a superpixel proposal, as illustrated in Fig. 3 (a) and (b) respectively. The connecting probability ρ_e^- of the competitive edge is thus defined as

$$\rho_e^- = \begin{cases} \exp\{\dfrac{||\Lambda_i \bigcap \Lambda_j||}{||\Lambda_i \bigcup \Lambda_j||}\}, & v_i, v_j \in V_S, \Lambda_i \bigcap \Lambda_j \neq \emptyset \\ 0, & v_i, v_j \in V_S, \Lambda_i \bigcap \Lambda_j = \emptyset \\ 1, & \Lambda_i \subset \Lambda_j \quad or \quad \Lambda_j \subset \Lambda_i \end{cases} \tag{4}$$

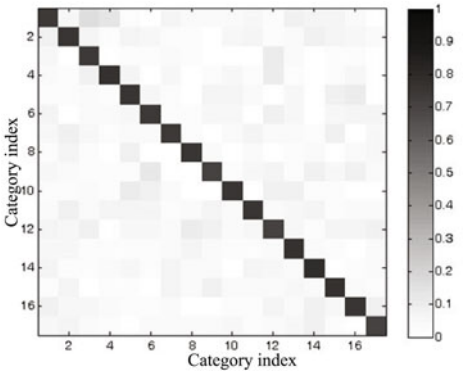


Fig. 4. Pairwise co-occurrence matrix for superpixel candidates

Contextual edges play a key role in our model, which imply contextual information between two graph vertices. We explore two types of contextual metrics: the co-occurrence between two superpixel candidates and the layout between a template candidate and an adjacent superpixel proposal.

- **Co-occurrence context**, constrains the two connected superpixel candidates according to a learned distribution of related object categories, as represented by a pairwise co-occurrence matrix $H^O(\cdot,\cdot)$ in Fig. 4. In this figure, the co-occurrence probabilities are scaled to gray-levels with the diagonals contributing zero energy. The darker gray-scale intensity denotes higher co-occurrence probability. For example, the "car" to "grass" pair has comparatively low probability because they seldom appear adjacently in our dataset. Conversely, the probability between "sky" and "mountain" is intuitively high as a result of their frequent co-occurrence in natural scene images.
- **Layout context**, constrains the relative location of a superpixel candidate, given a template candidate. For each structural category and surrounding superpixels in training set, we learn a 2D probability histogram $H^L_{l_i}(\cdot,\cdot)$ that encodes normalized pixel number with variational quadrant index and category index. Fig. 5 illustrates this layout context and the histogram.

Based on the definition of two context metrics, the probability of contextual edges ρ_e^+ is thus defined as,

$$\rho_e^+ = \begin{cases} H^O(l_i, l_j), & v_i, v_j \in V_T \\ \sum_{k=1}^{n(D)} \#(\Lambda_j \cap D_k) H^L_{l_i}(D_k, l_j), & v_i \in V_S \text{ and } v_j \in V_T \end{cases} \tag{5}$$

where D_k refers the pixel map of the j-th quadrant. $n(D)$ denotes the number of quadrant and we set it as 10. $H^L_{l_i}(\cdot,\cdot)$ is the layout context histogram with respect to template candidate v_i. Λ_j and l_j indicate image domain and semantic label of the superpixel candidate v_j respectively.

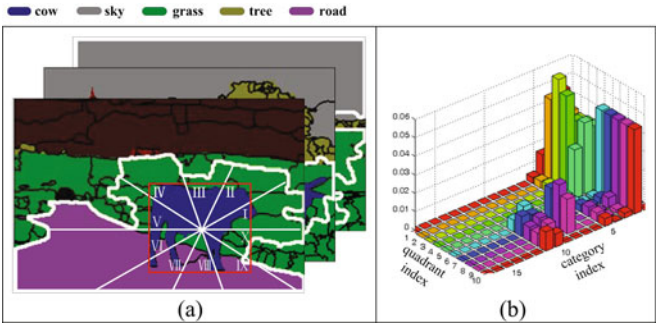


Fig. 5. Layout context probability histogram for the template candidates with surrounding superpixel candidates. (a) illustrates the definition of the layout context histogram. Given a structural object in training images, the annotated surrounding superpixels are projected to a number of quadrant (denoted by the Rome number). Thus the pixel number distribution over the quadrant index and category index can be calculated as shown in (b).

3 Probabilistic Formulation

Assume two sets of candidates are validated from the candidate set V: $Q = \{q_i\} \subset V_T, U = \{u_i\} \subset V_S$ with $\omega = 1$, and sizes of Q, U are N_T, N_S respectively. The solution configuration of parsing is defined as,

$$W = \left\{ N_T, Q = \{q_i\}_{i=1}^{N_T}, N_S, U = \{u_i\}_{i=1}^{N_S} \right\}, \tag{6}$$

We further formulate the solution W in Bayesian framework and solve it by maximizing a posterior probability as

$$W^* = \arg\max p(W|I) = \arg\max p(W)p(I|W). \tag{7}$$

Prior term. We define the prior probability over the candidate numbers N_T, N_S and the validating set $\Psi = (Q \bigcup U)$, as,

$$\begin{aligned} P(W) &= P(N_T)P(N_S)P(Q,U) \\ &\propto \exp\{-\alpha_T N_T\} \exp\{-\alpha_S N_S\} \cdot P(Q,U), \end{aligned} \tag{8}$$

where α_T and α_S are tuning parameters and are set as 1 empirically. Since Q and U are two types of candidates sharing the same definition as graph vertices, we define

$$P(Q,U) = \prod_{e \in E^+} \exp\{\beta \mathbf{1}(\omega_i = \omega_j)\} \prod_{e \in E^-} \exp\{\beta \mathbf{1}(\omega_i \neq \omega_j)\}. \tag{9}$$

$\beta \in [0,1]$ is the tuning parameter and it set as 0.5 in practice. $\mathbf{1}(\cdot) \in \{0,1\}$ is an indicator function for a Boolean variable. The probability is maximized when

all contextual edges have same vertices state and all competitive edges connect two vertices with differently state labels.

Likelihood term. Using the classifiers for the recognition candidate generation, we define the likelihood probability of our model with the validated proposals, as

$$P(I|W) = P(I|Q,U) \propto \prod_{q_i \in Q} \exp\{-E_T(q_i)\} \prod_{u_i \in U} \exp\{-E_S(u_i)\}, \quad (10)$$

where E_T and E_S are energy costs for each validated candidate given the two types of classifiers, as defined in Eq. 1 and Eq. 2.

4 Inference by Composite Cluster Sampling

Based on the candidacy graph $G =< V, E >$, our algorithm simulates a Markov chain that consist of a sequence of states in the solution space, and travels the space by realizing reversible jumps between any two successive states. For each stochastic jump step, whether a new state is accepted is decided by the Metropolis-Hastings [19] method that guarantees the global convergence of the inference algorithm. Given two successive states A and B, the acceptance rate is defined as,

$$\alpha(A \rightarrow B) = \{1, \frac{Q(B \rightarrow A)P(B)}{Q(A \rightarrow B)P(A)}\}, \quad (11)$$

where $P(A)$ and $P(B)$ are the posterior probability. $Q(B \rightarrow A)$ and $Q(A \rightarrow B)$ are canidae probability of "jumping" between two states. Following the theoretical analysis reported in [20], $Q(B \rightarrow A)/Q(A \rightarrow B)$ can be simplified by cluster sampling, which contains two steps: 1) forming a composite cluster, called connected component (CCP), by sampling the probabilistic edge connection; 2) flipping the generated CCP by re-validating graph vertices.

Forming a composite CCP in $G =< V, E >$ is equivalent to sampling the edge probability (defined in Eq. 4 and Eq. 5). For each probabilistic link $e =< v_i, v_j >$, we define the sampling protocol for edge sampling (cutting) as

- **Deterministic cut**, as illustrated by black "×" in Fig. 6, is performed (i) on contextual edges connecting two different state vertices, and (ii) on competitive edges connecting two same state vertices.
- **Probabilistic cut**, is illustrated by black "‖" in Fig. 6. (i) The contextual edges connecting two same state vertices are turn off with probability $1-\rho_e^+$, and (ii) the competitive edges connecting two different state vertices are turn off with probability $1-\rho_e^-$.

Note we then select one CCP with equal probability if more than one is formed. Thus the generation of the composite cluster can be calculated by the probability of "turning off" the edges (as the black "‖" and "×" denote in Fig. 6) around the composite cluster, as

$$Q(CCP) = \prod_{e \in E^+ \bigcap \mathcal{C}} (1-\rho_e^+) \prod_{e \in E^- \bigcap \mathcal{C}} (1-\rho_e^-), \quad (12)$$

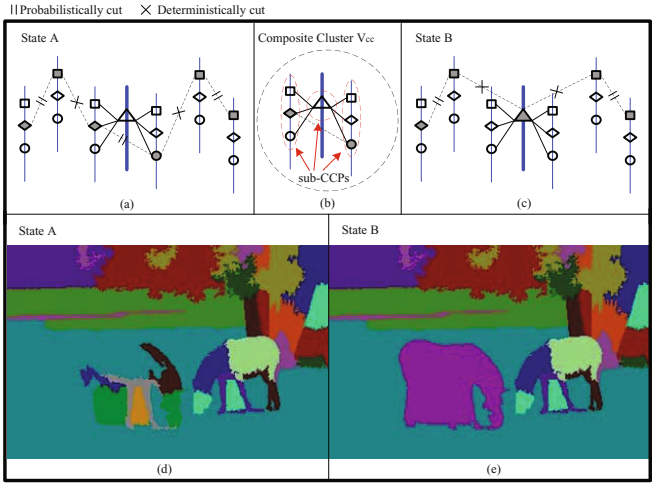


Fig. 6. The illustration of composite cluster sampling. (a) shows a current state and the edge links are cut deterministically or probabilistically denoting by black "×" or "‖"; (b) shows a composite cluster (in the black ellipse) including three conflicting connected components (CCPs) (in the dashed rectangles), and each one includes vertices connected by contextual (dashed) links while any neighboring CCPs are connected by a competitive (solid) link; (c) is the new state resulting from the re-validating in the composite cluster, and all vertices in a CCP are compatibly validated as a whole; (d) and (e) are the real segmentation solutions corresponding to states (a) and (c). Note the solid vertices imply validated candidates.

where $\mathcal{C}$ is a set of the edges that has been "turned off" around CCP. The edge probability ρ_e^+ and ρ_e^- are defined in Eq. 5 and Eq. 4 respectively. Note we then u.a.r select one CCP if more than one is formed.

Flipping the CCP is equivalent to re-validating vertices in the CCP. We split the selected CCP to many sub-CCPs, as showed in Fig. 6 (b), and then simply reverse the state of each vertex thus keeping the current constraints satisfied, since our candidacy representation is a typical Ising model where each site only has two states.

Thus $Q(B \to A)/Q(A \to B)$ only depends on the generation of CCP, and computed by

$$\frac{\prod_{e\in E^+ \bigcap \mathcal{C}_B}(1-\rho_e^+)\prod_{e\in E^- \bigcap \mathcal{C}_B}(1-\rho_e^-)}{\prod_{e\in E^+ \bigcap \mathcal{C}_A}(1-\rho_e^+)\prod_{e\in E^- \bigcap \mathcal{C}_A}(1-\rho_e^-)}. \tag{13}$$

A representative composite cluster CPP with three conflicting connected components (sub-CCPs) is shown in Fig. 6 (b), and all vertices in each sub-CCP should be compatibly validated with the other neighboring ones. Fig. 6 (c) and (d) demonstrate the real segmentation solutions corresponding in a step of reversible jump, taking fully advantage of the composite cluster.

The overall description for this composite cluster sampling algorithm is summarized in Algorithm 1.

Algorithm 1. Inference Algorithm

Input: testing image I, superpixel-based candidate set V_T , template-based candidate set V_S
Output: convergence solution $W^* \sim P(W|I)$

Construct graph representation: $G =< V, E >$.
repeat to sample loop for W to get the final solution
  begin Cut edges to form CCP sets $\{V_i, i = 1, 2, \ldots, M\}$ by edge strength
    for *each contextual edge* **do**
      if *two vertices have the same state* **then**
        cut the edge by probability $1 - \rho_e^+$
      else
        cut the edge deterministically
    for *each competitive edge* **do**
      if *two vertices have the same state* **then**
        cut the edge deterministically
      else
        cut the edge by probability $1 - \rho_e^-$
  end
  Randomly select a composite CCP;
  Revalidated each sub-CCP of CCP to form a new state W';
  Calculate the accept probability $\alpha(W \rightarrow W')$ by Eq. 11 to move to next solution or not.
until *Predefined criterion is satisfied.*;

5 Experiments

We evaluate our approach on two public data sets: (i) MSRC 21-class database [3] that contains 591 images in total, and (ii) LHI 17-class database [21] including $17 \times 15 = 255$ images. Compared to the LHI database, the MSRC database was published earlier and many researchers reported result on it, however its groundtruth annotation is relatively rough as a benchmark for semantic parsing task. For both data sets, the images are normalized into size of 320×213 and all images are split randomly into roughly 45% for training, 10% for validation and 45% for testing as well as [3] does. The algorithm is implemented by C++ on a PC with Core Duo 2.8 GHZ CPU. The computation cost is comparatively lower and it spends around $40 \sim 60s$ per image. The average sampling cost time for convergence is only around $4 \sim 6$ seconds based on the generated candidates. The speed reported in [3] was 3 minutes and 70 seconds in [8].

MSRC 21-calss databae. We randomly split the dataset into 337 images for training and 254 ones for testing, like in [3]. Some typical results are shown in Fig. 7, and the quantitative overall pixel-wise accuracy with comparison is reported in Table 1.

Table 1. Overall pixel-wise accuracy on MSRC 21-class database [3] and LHI 17-class database

Methods	MSRC	LHI
Proposed	77.6%	77.2%
CRF + Rel.Loc. [4]	76.5%	N/A
Bag of Keypoints [24]	75.1%	N/A
Auto-Context [8]	72.9%	N/A
TextonBoost [3]	72.2%	67.2%
Geodesic-distance [22]	N/A	71.4%

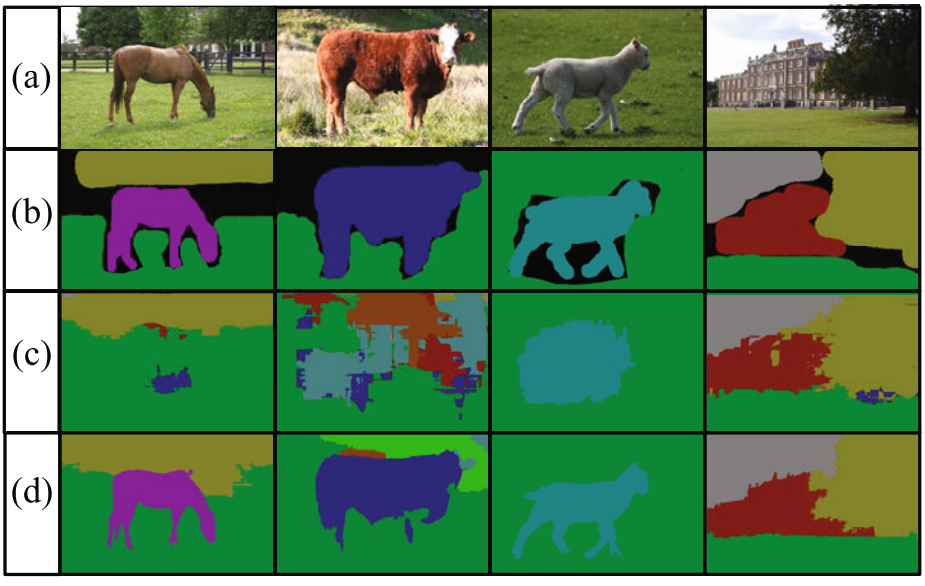


Fig. 7. A few typical results on MSRC 21-class data set. From the top row to the bottom row are: original images, annotated label maps, Result by [3], and our results. The different color denotes the different object category. The overall pixel-wise accuracy is proposed in Table. 1.

LHI 17-class database. We further test our approach on more challenging LHI database that provides more accurate annotated groundtruth. A number of original images, annotation labeling, superpixel-based candidates, template-based candidates, and the final results are presented in Fig. 9. A few examples of iterative sampling are exhibited in Fig. 10(b). The confusion matrix of multi-class recognition for total 17 categories is proposed in Fig. 8, and the overall pixel-wise accuracy on this dataset is 77.2%. The TextonBoost [3] on this dataset outputs 67.2%.

In another comparison, we implement a recently presented geodesic-distance method [22] to achieve segmentation based on our superpixel-based candidates. Like the graph cuts [9], geodesic-distance algorithm deterministically assigns

	Bu	Gs	Tr	Sk	Mt	W	Cr	Rd	Bt	Sd	Gd	Cw	Sp	Ct	Hr	Dg	Rh
Building	**0.8275**	0.0197	0.002	0.0254	0.0124	0.0185	0.018	0.0444	0.0028	0.0031	0.0029	0.0059	0.0038	0.0027	0.0037	0.0043	0.0029
Grass	0.0099	**0.8829**	0.0057	0.0031	0.0021	0.0061	0.0041	0.007	0.009	0.0126	0.009	0.0009	0.0012	0.0099	0.012	0.0128	0.0118
Tree	0.0173	0.0448	**0.7654**	0.0214	0.0129	0.0189	0.0205	0.0135	0.0076	0.0094	0.0083	0.0083	0.0133	0.0088	0.0111	0.0095	0.0092
Sky	0.002	0.0034	0.0047	**0.9597**	0.009	0.0048	0.0013	0.0012	0.0012	0.0013	0.0012	0.003	0.0012	0.0013	0.002	0.0013	0.0012
Moutain	0.0095	0.0159	0.0154	0.12	**0.7999**	0.0107	0.0064	0.0023	0.0017	0.0024	0.0021	0.0015	0.0067	0.0015	0.0015	0.0014	0.0014
Water	0.0106	0.0441	0.0329	0.0322	0.0214	**0.7527**	0.0184	0.0131	0.009	0.0127	0.0053	0.0054	0.0053	0.0053	0.0133	0.0108	0.0075
Car	0.0104	0.0169	0.0221	0.0187	0.0143	0.0161	**0.7836**	0.0558	0.0067	0.0088	0.0066	0.0068	0.0065	0.0066	0.0067	0.0067	0.0067
Road	0.009	0.0351	0.0097	0.0077	0.0087	0.0108	0.0379	**0.8068**	0.0077	0.0076	0.0072	0.0085	0.0086	0.008	0.0073	0.0119	0.0075
Boat	0.0162	0.0097	0.0092	0.0257	0.0354	0.023	0.0098	0.0102	**0.7756**	0.02	0.009	0.0096	0.0093	0.0093	0.009	0.0093	0.0096
Sand	0.0144	0.0071	0.0174	0.023	0.0146	0.0057	0.018	0.0056	0.0147	**0.7636**	0.0145	0.0144	0.0153	0.0182	0.0183	0.0203	0.0151
Ground			0.0006	0.0703	0.0017						**0.9244**				0.003		
Cow	0.0109	0.0174	0.0287	0.0349	0.0126	0.0148	0.0111	0.0283		0.011	0.0112	**0.7442**	0.0112	0.0112	0.016	0.0143	0.011
Sheep	0.0024	0.0428	0.0705	0.0034	0.0519	0.002	0.0001	0.0355		0.0074			**0.7839**				
Cat		0.1155	0.0835			0.0068		0.0788		0.0089				**0.7064**			
Horse	0.01	0.1399	0.0511	0.0662	0.0267	0.0381	0.0161	0.0162	0.0164	0.026	0.0292	0.0159	0.0162	0.0162	**0.5355**	0.0183	0.0159
Dog	0.0372	0.1462	0.0474	0.0213	0.0075	0.0697	0.0088	0.0372	0.0061	0.0079	0.0059	0.0072	0.0061	0.0065	0.0065	**0.5726**	0.006
Rhinoceros	0.0074	0.1074	0.0287	0.0076	0.0074	0.0074	0.0139	0.0145	0.0075	0.018	0.0073	0.0078	0.0074	0.0076	0.0075	0.0078	**0.7348**

Fig. 8. Confusion matrix of labeling for total 17 categories on the LHI 17-class [21] database. The overall accuracy is 77.2%.

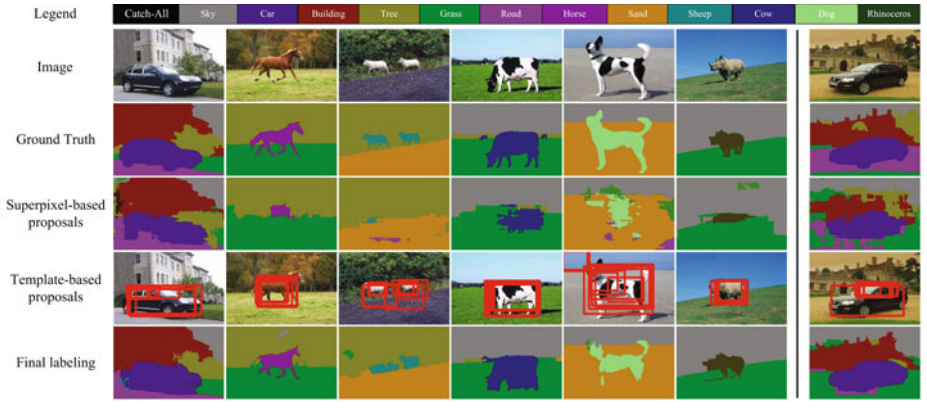


Fig. 9. Example results on LHI 17-calss database [21]. We demonstrate a few original images, annotation labeling, superpixel-based candidates, template-based candidates, and the final results from the top row to the bottom row. The column on the right is a failure example.

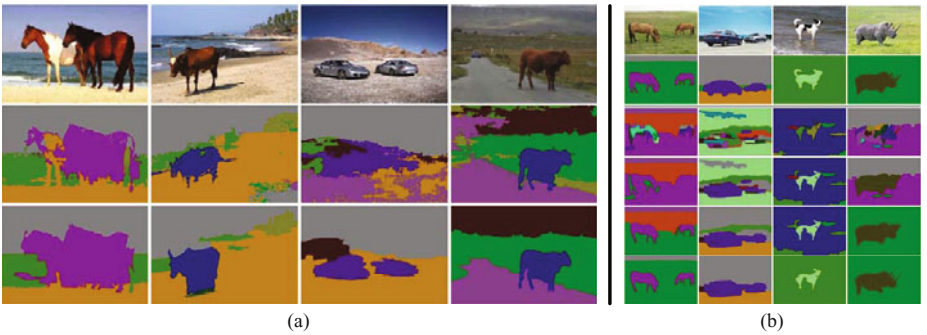


Fig. 10. (a)A comparison with geodesic-distance method (in the middle row). The original images and our results are shown in the top row and the bottom row respectively. (b)An illustrative examples of iterative sampling. The top two rows exhibit the original images and the groundtruth annotation. The other rows (3-rd $\sim$ 6-th) show the results in iterative sampling.

label to each pixel based on confident initialization. In this experiment, we set the initialization by a few candidates with low energy cost. The overall accuracy reaches 71.4%. However, this deterministic algorithm often depends on confident initialization and may stuck in local minimal. Three typical examples of geodesic-distance method are exhibited in Fig. 10(a) to compare with our method.

6 Summary

For the semantic scene understanding task, this paper studies a candidacy graphical representation of integrating the textural appearance and shape information. In contrast to the current methods using textural appearance and pixel-level context information, we additionally explore the object structural model in the candidacy representation, as well as the competitive and contextual interactions. An efficient composite sampling algorithm based on this representation is proposed in the Bayesian framework. Unlike the traditional single-site sampler, this algorithm updates large portions of the solution space quickly to minimize constraint energy, by clustering connected components in each sampling step. Our approach is test on both LHI and MSRC public data sets and outperforms the state-of-art methods.

Acknowledgement. We also thank Jun Zhu, Tianfu Wu, Lin Liang, Xiang Bai, and Yu Zhou for their helpful discussions. This work was supported by NSFC No.60873127, 60903096 and 60903172.

References

1. Laferte, J.M., Heitz, F., Perez, P., Fabre, E.: Hierarchical statistical methord for the fusion of multiresolution data. In: ICCV (1995)
2. Xuming, H., Zemel, R.S., Carreira-Perpinan, M.A.: Multiscale conditional random fields for image labeling. In: CVPR (2004)
3. Shotton, J., Winn, J., Rother, C., Criminisi, A.: Textonboost: Joint appearnce, shape and context modeling for multiclass object recognition and segmentation. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 1–15. Springer, Heidelberg (2006)
4. Gould, S., Rodgers, J., Cohen, D., Elidan, D., Koller, D.: Multi-class segmentation with relative location prior. IJCV 80, 300–316 (2008)
5. Galleguillos, C., Rabinovich, A., Belongie, S.: Object categorization using co-occurrence, location and appearance. In: CVPR (2008)
6. Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., Belongie, S.: Object in context. In: ICCV (2007)
7. Bastian, L., Ales, L., Bernt, S.: Combined object categorization and segmentation with an implicit shape model. In: ECCV Workshop on Statistical Learning in Computer Vision (2004)
8. Zuowen, T.: Auto-context and its application for high-level vision tasks. In: CVPR (2008)
9. Kolmogorov, V., Zabin, R.: What energy functions can be minimized via graph cuts? PAMI 26, 147–159 (2004)

10. Frey, B.J., Mackay, D.: A revolution: Belief propagation in graphs with cycles. In: NIPS (1997)
11. Apt, K.: The essence of constraint propagation. Theoretical Computer Science 221, 179–210 (1999)
12. Geman, S., Geman, D.: Stochastic relaxation, gibbs distributions and the bayesian restoration of images. PAMI 6, 721–741 (1984)
13. Ferrari, V., Jurie, F., Schmid, C.: Accurate object detection with deformable shape models learnt from images. In: CVPR (2007)
14. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. PAMI 24, 509–522 (2002)
15. Yinnian, W., Zhangzhang, S., Songchun, Z.: Deformable template as active basis. In: ICCV (2007)
16. Xiang, B., Xinggan, W., Longin, J.L., Wenyu, L., Zuowen, T.: Active Skeleton for Non-rigid Object Detection. In: ICCV (2009)
17. Borenstein, E., Ullman, S.: Combined top-down/bottom-up segmentation. PAMI 30, 2109–2125 (2008)
18. Tu, Z.W., Chen, X., Yulle, A., Zhu, S.: Image parsing: Unifying segmentation, detection, and recognition. IJCV 63 (2005)
19. Metropolis, N.: Equation of state calculations by fast computing machines. Journal of Chemical Physics 21, 1087–1092 (1953)
20. Barbu, A., Zhu, S.: Generalizing swendsen-wang for image analysis. Journal of Computational and Graphical Statistics 16, 877–900 (2007)
21. Yao, B., Yang, X., Zhu, S.-C.: Introduction to a large-scale general purpose ground truth database: Methodology, annotation tool and benchmarks. In: Yuille, A.L., Zhu, S.-C., Cremers, D., Wang, Y. (eds.) EMMCVPR 2007. LNCS, vol. 4679, pp. 169–183. Springer, Heidelberg (2007)
22. Bai, X., Sapiro, G.: Geodesic matting: A framework for fast interactive image and video segmentation and matting. IJCV 82, 113–132 (2009)
23. Torralba, A., Murphy, K.P., Freeman, W.T.: Sharing features: efficient boosting procedures for multiclass object detection. In: CVPR (2004)
24. Lin, Y., Meer, P., Foran, D.J.: Multiple class segmentation using a unified framework over mean-shift patches. In: CVPR (2007)

Saliency Density Maximization for Object Detection and Localization

Ye Luo[1], Junsong Yuan[1], Ping Xue[1], and Qi Tian[2]

[1] School of EEE, Nanyang Technological University, Singapore
[2] Institute for Infocomm Research, Singapore

Abstract. Accurate localization of the salient object from an image is a difficult problem when the saliency map is noisy and incomplete. A fast approach to detect salient objects from images is proposed in this paper. To well balance the size of the object and the saliency it contains, the salient object detection is first formulated with the maximum saliency density on the saliency map. To obtain the global optimal solution, a branch-and-bound search algorithm is developed to speed up the detection process. Without any prior knowledge provided, the proposed method can effectively and efficiently detect salient objects from images. Extensive results on different types of saliency maps with a public dataset of five thousand images show the advantages of our approach as compared to some state-of-the-art methods.

1 Introduction

Detection of the salient object from an image has many applications in object recognition [1], image/video retargeting [2], compression [3], retrieval etc. To find the salient object, a saliency map of the image is firstly generated, where each pixel is associated with a value that indicates the importance of the pixel. Then the salient object can be detected or segmented from the saliency map.

A lot of efforts have been reported in saliency detection. However, accurate localization of the salient region or salient object from an image is still a challenging and non-trivial problem. First of all, it is not uncommon that the obtained saliency map is noisy and incomplete. As shown in Fig. 1, only several salient parts of the flower are highlighted, while the rest are missing. Due to the distraction from the cluttered background, it is not easy to find the salient region and accurately crop it out. Moreover, most existing methods apply exhaustive search for the smallest region that covers a fixed amount of fixation points [4,5,6], e.g. 95 % of the total salient points [4]. The major limitation is that it is difficult to predefine the amount of saliency the salient region should contain, as it depends on the size and shape of the salient object, as well as how cluttered the background is. Ideally, the salient region should be adapted to the shape of the salient object.

To address the mentioned problems, we propose a novel method to efficiently detect salient object from the saliency map. Given the saliency map, the goal is to locate a bounding box from the image that has a small size and contains most of

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 396–408, 2011.

Fig. 1. Our salient object detection result based on the saliency map proposed in [7](The first image is the saliency map by [7]. The second image is the binarized result of the saliency map. The third image is the result of salient object detection by searching method in [8]. The red bold rectangle is the detected result while the blue plain one is the ground truth from [4]. The last one is the result by our MSD detection method.).

the salient parts of the image. We formulate the problem as region localization with the maximum saliency density (MSD). As a new formulation of salient object detection, it balances the size of the object and the saliency it contains, and can tolerate the noise and incompleteness in the saliency map. As shown in Fig. 1, even though the salient pixels distribute sparsely, the detected saliency region with highest saliency density accurately crops the object out. Our method does not require any prior knowledge of the salient object and can automatically adapt to its size and shape through bounding box search. To avoid an exhaustive search of all possible bounding boxes of various sizes and at different locations, a branch-and-bound (B&B) search algorithm is proposed to efficiently find the global optimal bounding box.

There are several advantages of our method. First of all, it can automatically adapt to the size and shape of the salient object, despite the cluttered background. There is no need to find salient object with fixed fraction of saliency and it does not require the binary mask of the saliency map, where pixels need to be classified into salient and non-salient ones. Instead, it directly finds the bounding box of maximum saliency density from the original saliency map. Moreover, by using the branch-and-bound search, it is fast to find the optimal bounding box, e.g. in tens of milliseconds. Last but not least, our new formulation and search algorithm can be well applied to different types of saliency maps. A better performance is achieved when performed on a fused map of different types of saliency maps.

2 Related Work

2.1 Saliency Map

Literally, there are two categories of computational saliency map models: local or edge/corner based [7,8,9,10] and global or region based [4,11,12]. The first

row in Fig. 3 shows several examples. The 1^{st} and 4^{th} columns are based on Hou's method [7]. The 3^{rd} and 6^{th} columns are from Bruce's method [9]. And the 2^{nd} and 5^{th} columns are from Achanta's [12], which generates larger visually consistent object regions than that of the previous two. In this paper, saliency map generation method is not our focus. Our main work is to detect salient objects from various saliency maps.

2.2 From Saliency Map to Salient Object

The simplest method to obtain the salient object region is by thresholding the saliency map to get a binary mask. Methods to threshold saliency map are intensively discussed in [5,12,13,14]. This method is restricted on the selection of threshold and detection accuracy. In order to accurately detect salient objects from saliency maps, image segmentation result is combined with the saliency map [10,8,12]. However, the performance heavily relies on the accuracy of image segmentation results. Some heuristic methods [4,15,16,17] are proposed to improve the performance of salient object detection. For example, exhaustive search is adopted in [4] to find the smallest rectangle window containing 95% salient pixels. Liu et al. [5] noticed the disadvantages of exhaustive search and proposed to use dynamic threshold and greedy algorithm to improve the search efficiency. However, their method is still based on thresholds and not sovled by a standard optimization method. In [8], the search of the rectangle subwindow is speeded up by applying the efficient subwindow search (ESS). ESS is a recently proposed branch-and-bound search method for sliding window search [18]. It has many applications in image/video analysis [8,18,19].

3 The Proposed Method

Given an image I and its associated saliency map S, where $S(x, y)$ indicates the saliency value of the pixel at (x, y), our goal is to accurately locate the salient object, i.e. to locate a salient region $W \subseteq I$. We first review existing methods then propose our new approach.

3.1 Existing Schemes

Exhaustive search (ES). Some previous approaches proposed to obtain salient object regions with fixed fraction of saliency by exhaustive search from saliency maps [5,12] or binary saliency maps [4]. We take the binary saliency map as in [4] and the formulation can be written as in Eq. 1:

$$W^* = \arg\min_{W \subseteq I} area(h(W)) \tag{1}$$
$$h(W) = \{W | \sum_{(x,y)\in W} S_b(x, y) \geq \lambda \sum_{(x,y)\in I} S_b(x, y)\}.$$

where $S_b(x, y)$ is the binary image of $S(x, y)$. $S_b(x, y) = 1$ when $S(x, y) \geq \tau$ and $S_b(x, y) = 0$ when $S(x, y) \leq \tau$. τ is the threshold and W is the subwindow of

the whole image region I. λ is the fraction threshold. The brute force method works, however, it is not time efficient and λ is heuristically decided.

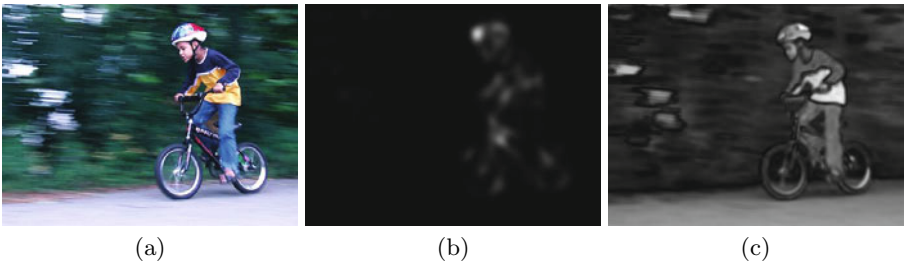
Maximum saliency region (MSR). Other approaches proposed to detect the salient object by efficient subwindow search [8]. Since efficient subwindow search is based on the maximum subarray problem [18,20], the idea of salient object detection in [8] can be formulated as in Eq. 2:

$$W^* = \arg\max_{W \subseteq I} h(W) \tag{2}$$
$$h(W) = \sum\nolimits_{S_b(x,y) \in W} S_b(x,y).$$

where $S_b(x,y)$ is obtained in the same way in Eq. 1 with a slight difference that $S_b(x,y) = -1$ when $S(x,y) \leq \tau$. From Eq. 2, the salient object is located with the region W^* that contains the maximum of saliency. We call this method as the maximum saliency region (**MSR**). However, there are two major limitations of this method: (1) it highly relies on the selection of threshold τ, which is difficult to optimize; (2) when the binary saliency map is sparse, it prefers to detect a small region as shown in Fig. 1.

3.2 Our New Formulation

Before giving our new formulation, we first introduce the concept of sparse and dense saliency map. Fig. 2 shows two examples. Since different saliency map generation method emphasizes different aspect, edges or corners of the salient object in Fig. 2(b) are highlighted while in Fig. 2(c) the whole salient object is popped out with uniform highlighted intensities. Therefore, we can say that the salient object in Fig. 2(b) is sparsely represented by Hou's saliency map and it is densely represented by Achanta's saliency map in Fig. 2(c). Sparse saliency map accurately detects the salient parts of the object but the boundary of the salient object is not well defined. Dense saliency map represents the salient object completely but some cluttered background is also included in the detection result. However, one thing is in common: the averaged density of the salient object region is much larger than that of any other regions on the saliency map.



(a) (b) (c)

Fig. 2. (a)Original image (b) Sparse saliency map example by [7] (c) Dense saliency map example by [12]

To address the above characteristics and the problems in section 3.1, we propose to find the salient region W^* with the maximum saliency density from the raw saliency map $S(x, y)$. Thus we do not need to select the threshold τ and the fraction ratio λ. Moreover, it balances the size of the salient object when the saliency map is sparse. We formulate our objective function $f(W)$ as:

$$W^* = \arg\max_{W \subseteq I} f(W) \tag{3}$$
$$f(W) = \frac{\sum_{(x,y)\in W} S(x,y)}{\sum_{(x,y)\in I} S(x,y)} + \frac{\sum_{(x,y)\in W} S(x,y)}{C + Area(W)}.$$

where C is a positive constant to balance the size of $Area(W)$. The first term in $f(W)$ prefers that W contains more salient points, while the second term ensures that the detected region W is of high quality in terms of the saliency density. Therefore, by maximizing the two terms together in $f(W)$, we balance the size of the object and the saliency it contains. We call our new formulation as the maximum saliency density (**MSD**).

4 Our Algorithm

Exhaustive search of W^* from Eq. 3 is time consuming. $W^* = [T, B, L, R]$ contains four parameters, where T, B, L, R are the top, bottom, left, and right position of W^*, respectively. Suppose the frame is of size $m \times n$, the original hypotheses space is $[0, n-1] \times [0, n-1] \times [0, m-1] \times [0, m-1]$, where we need to pick up T, B, L, R from each dimension respectively. To solve this combinatorial problem, an exhaustive search is of complexity $O(m^2n^2)$. A branch-and-bound search method is proposed in [18] to accelerate the search by recursively partitioning the parameter space until it reaches the optimal solution. It shows that under certain conditions, such a branch-and-bound search can lead to the exact solution as the exhaustive search, while with a practical complexity of only $O(mn)$. The details of the branch-and-bound search can be referred to [18].

The original branch-and-bound only works for the saliency map having both positive and negative pixel values. However, in our case, the saliency map only contains positive elements and we do not want to deliberately introduce negative pixels. Therefore we need to derive our own branch-and-bound search method. Considering the efficiency of branch-and-bound search depends on the upper bound estimation, we derive the upper bound of our $f(W)$ first. Denote the set of regions by $\mathbb{W} = \{W_1, \ldots, W_i\}$, where each $W_i \subseteq I$. Suppose there exists two regions W_{min} ($W_{min} \in \mathbb{W}$) and W_{max} ($W_{max} \in \mathbb{W}$), such that for any ($W \in \mathbb{W}$), $W_{min} \subseteq W \subseteq W_{max}$. Given the set $\mathbb{W}$, we denote by $\hat{f}(\mathbb{W})$ the upper bound estimation of the best solution that can find from $\mathbb{W}$. In other words, we have $\hat{f}(\mathbb{W}) \geq f(W)$, $\forall W \in \mathbb{W}$, using W_{min} and W_{max}, the upper bound can be estimated as:

$$\hat{f}(\mathbb{W}) = \frac{\sum_{(x,y)\in W_{max}} S(x,y)}{\sum_{(x,y)\in I} S(x,y)} + \frac{\sum_{(x,y)\in W_{max}} S(x,y)}{C + Area(W_{min})}. \tag{4}$$

Based on this upper bound estimation, we propose our MSD salient object detection algorithm as shown in table 1, in which the branch-and-bound procedure is similar to that of [18].

Table 1. MSD Salient Object Detection Algorithm

Require:	Image saliency map $S \subseteq R^{m \times n}$
	Upperbound function $\hat{f}(\mathbb{W})$ as Eq. 4
Ensure:	$W^* = \arg\max_{W \subseteq I} f(W)$
	Initialize P as an empty priority queue
	Set $\mathbb{W} = [0, n-1] \times [0, n-1] \times [0, m-1] \times [0, m-1]$
	Repeat
	Split $\mathbb{W} = \mathbb{W}_1 \cup \mathbb{W}_2$ and $\mathbb{W}_1 \cap \mathbb{W}_2 = \emptyset$
	For i = 1 to 2
	Find W_i^{min} and W_i^{max} from $\mathbb{W}_i$
	Push $(\mathbb{W}_i, \hat{f}(\mathbb{W}_i)$ into P
	End For
	Retrieve top state $\mathbb{W}$ from P
	Until $\mathbb{W}$ contains only one window, e.g. $W^{min} = W^{max}$
	Return $W^* = W^{min}$

5 Experimental Results

5.1 Database

In order to evaluate the results, a public dataset [4] is used to test our algorithm. The dataset provides 5000 high quality images each of which contains a salient object. Each salient object is labeled by nine users by drawing a bounding box around it. Since different users have different understanding of saliency, the point voted more than four times is considered as the salient point. The averaged saliency map S_g is then obtained from user annotation.

Given the ground truth S_g which is the binary mask of the salient object, we evaluate the performance of our method based on precision, recall, and F-measure. Suppose S_d is the salient region found by our method, the precision, recall and F-measure can be defined as:

$$pre = \frac{\sum S_g \times S_d}{\sum S_d}, rec = \frac{\sum S_g \times S_d}{\sum S_g}, F - measure = \frac{(1+\alpha) \times pre \times rec}{\alpha \times pre + rec}. \quad (5)$$

where α is a positive constant which weights the precision over recall while calculates F-measure. We take $\alpha = 0.5$ as suggested in [8,4].

5.2 Comparison MSD with Exhaustive Search

First of all, we compare our method with the exhaustive search. λ is set to 95% as [4] suggested. Fig. 3 shows the results obtained by the exhaustive search and

Fig. 3. Detection results comparison among our MSD, exhaustive search and MSR. The first row are two examples of saliency maps for Hou's [7], Achanta's [12] and Bruce's [9] methods respectively. The second row are localization results by exhaustive search on the three saliency maps. The third row are results by MSR. And the last row are our MSD results. Detected results are labeled with bold red line. Blue plain rectangles are ground truth from [4].

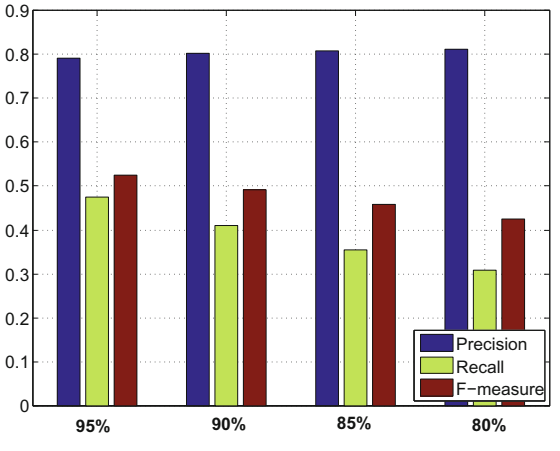
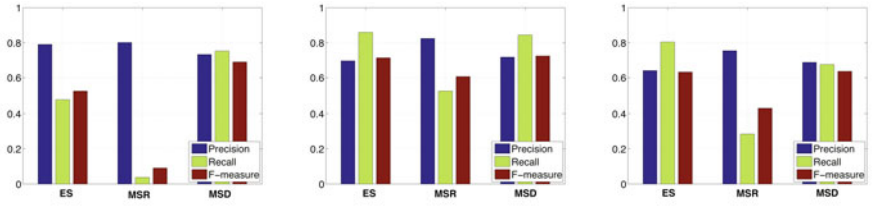


Fig. 4. Precision, recall and F-measure for exhaustive search with four λ {95%, 90%, 85%, 80%} on Hou's saliency map

our method on the second and the last rows respectively. As 95% is an arbitary value and not decided based on the content of the saliency map, the detected results include a large part of the nonsalient object area. While, in our method, small salient area away from main salient object region is dropped under the constraint of the saliency density. Therefore, our result is more accurate than



(a) (b) (c)

Fig. 5. Comparisons precision, recall and F-measure for our MSD to exhaustive search (ES) and MSR on (a) Hou's saliency map, (b) Bruce's saliency map and (c) Achanta's saliency map

the exhaustive search. Performaces of precision, recall and F-measure on Fig. 5 further validate our claim. In order to balance the bias caused by arbitarily choosing λ, we test four different λ values as Fig. 4 shows. Precision is improved while recall is reduced as λ becomes smaller and there is no direct way to choose an optimal λ.

5.3 Comparison MSD with MSR

To compare our MSD with MSR, we test both of the methods on three different saliency maps. The threshold τ is obtained by Otsu [22] for all saliency maps in MSR. C is set to be 60625, 2025 and 16200 for Hou's, Bruce's and Achanta's saliency map respectively in MSD, through parameter evaluation. Fig. 5 shows the comparison results. The average of precision, recall and F-measure are reported on each group. On Hou's saliency map, edges/corners are detected as salient parts. By using MSR, very small region is bounded while larger salient regions are detected by our MSD. The F-measure and recall are significantly improved by our MSD. For the other two saliency maps, our MSD also outperforms MSR. The results on three different types of saliency maps show that our method improves the F-measure, and at the same time, keeps the high precision rate.

5.4 Evaluation on Different Saliency Maps

Since the salient object detection result is based on the saliency map, the more accurate the saliency detection is the better performance of the object detection method obtains. It is worth noting that a single salient object region in [4] is obtained through supervised learning. Both [10] and [8] have prior knowledge about the region size provided by image segmentation. Thus, they are not directly comparable with our method. However, even without any prior knowledge of the salient object, our method on Bruce's saliency map outperforms Ma's method [16] which directly uses detected salient region (F-measure 61%) and search result on Itti's saliency map [21] which finds the minimum rectangle containing 95% salient points by the exhaustive search (F-measure 69%). For our method on Hou's saliency map, it obtains comparable result compared with

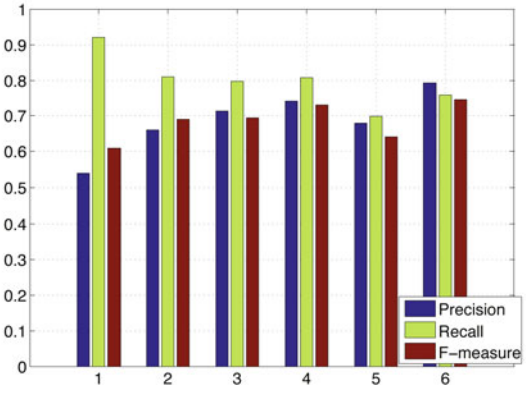


Fig. 6. Comparison our MSD to other salient object detection results by precision, recall and F-measure. 1: Ma's saliency map and their salient region detection result [16]; 2: Exhaustive search smallest subwindow containing 95% salient points from Itti's saliency map [21]; 3: MSD on Hou's saliency map; 4: MSD on Bruce's saliency map; 5: MSD on Achanta's saliency map; 6: MSD on the combined saliency map.

the exhaustive search on Itti's saliency map. Though our result on Achanta's saliency map is not as good as the result on [21], the precision is still higher than searching result on Itti's and Ma's saliency maps.

Since different bottom-up saliency map generation method has different advantages and disadvantages, to minimize the influence to the saliency object detection result, three previous saliency maps are fused together. Each saliency map is normalized into $[0, 1]$ and the combination saliency map is obtained by adding them together then normalizing the summation into $[0, 1]$. As shown in Fig. 6 method 6, after combining three saliency maps together, F-measure is 74.67% which is 1.61% larger than the optimal result (73.06%) from Bruce's saliency map. This performance is comparable to the learning based salient object detection results e.g. [4,8] but our method is much simple and time efficient than them.

5.5 Parameter Evaluation

To evaluate the influence of the only parameter C in our MSD method, different values C are tested as shown in Fig. 7. When C is small, the method is sensitive to the density change and prone to converge to a region with higher average density but relative smaller size. When a large value of C is selected, density term becomes trivial in objective function $f(W)$ and the whole algorithm converges to a larger region with lower average density. In Fig. 7(a), within the range of $[35000, 84000]$, it is thus not sensitive to the selection of C. Similarly, when C is in the range $[1500, 3300]$, the F-measure is above 71.3% in Fig. 7(b); when C in the range $[14200, 23500]$, the F-measure is above 63.5% in Fig. 7(c). From these results, we can see that the region based saliency map has a smaller optimal C

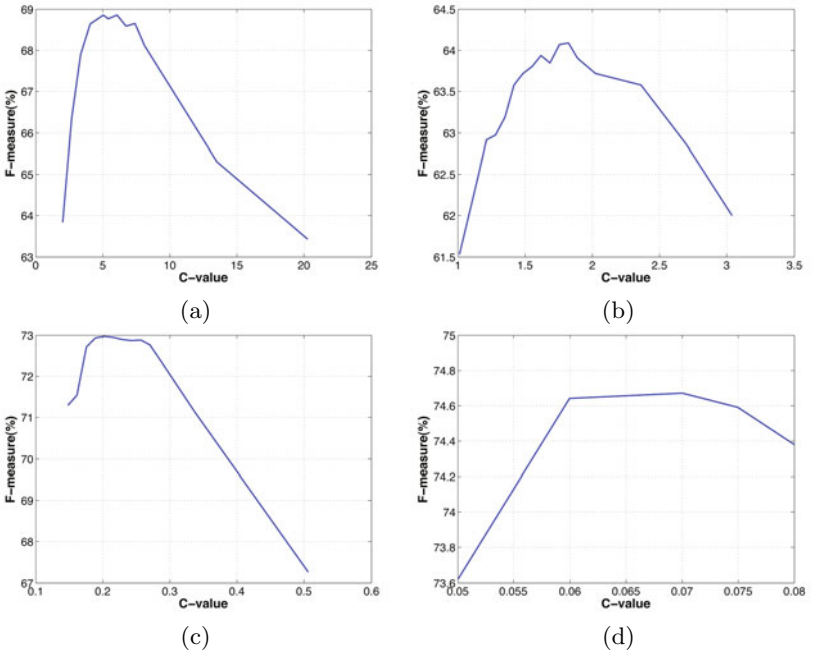


Fig. 7. Optimal C-value range for MSD on (a) Hou's saliency map, (b) Bruce's saliency map, (c) Achanta's saliency map and (d) Combined saliency map (x-coordinate is C-value measured in unit 10^4 and y-coordinate is the corresponding averaged F-measure)

Table 2. Time complexity comparison by seconds

Method	saliency map by [7]	saliency map by [9]	saliency map by [12]
Exhaustive Search	22.2359	22.1646	23.3587
MSR	0.0039	0.0048	0.0056
Our MSD	0.0113	0.0351	3.4718

than edge/corner based methods (As Fig. 3 shows Bruce's saliency map is denser than Achanta's.). That further indicates that density term in Eq. 3 is important when the salient points are densely distributed on the saliency map.

5.6 Time Complexity

The average computational time tested on 5000 images for exhaustive search, MSR and our method based on Hou's, Bruce's and Achanta's saliency map are shown in table 2. It is obvious from table 2 that our method is very time efficient compared to the exhaustive search and has comparable time efficiency to

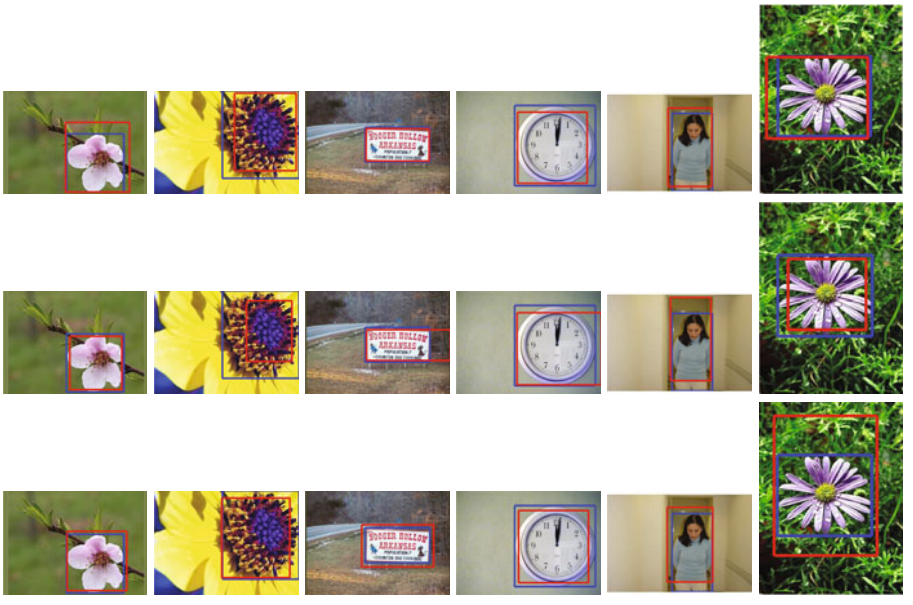


Fig. 8. More salient object localization results by our MSD method (The first row are our results on Hou's saliency map [7]. The second row are our results on Achanta's saliency map [12] and the last row are based on Bruce's saliency map [9]. The red bold rectangle is the detected result while the blue plain one is the ground truth from [4]).

MSR. The algorithm is tested on a Duo Core desktop of 2.66GHz, implemented with C++.

6 Conclusion

We propose in this paper a novel method to efficiently detect salient objects from images. Salient object detection is first formulated with the saliency density. A branch-and-bound search algorithm is developed to optimize the newly formulated problem globally. Without a prior knowledge of the salient object, our method can adapt to different sizes and shapes of the object, and is less sensitive to the cluttered background. The experiments on a public dataset of 5000 images show that our method greatly improves the existing baseline methods on the measurements of precision, recall and F-measure. Our method gains comparable performance compared to learning based salient object detection results with a high time efficiency. Tests on different saliency maps indicate our method works well with different types of saliency maps. Our future work includes the localization of multiple salient objects in images and content based image and video retargeting.

References

1. Gao, D., Vasoncelos, N.: Descriminant saliency for visual recognition from cluttered scenes. In: Advances in Neural Information Processing Systems, pp. 481–488 (2004)
2. Liu, F., Gleicher, M.: Video retargeting: Automating pan and scan. In: Proc. ACM Multimedia, pp. 241–250 (2006)
3. Bradley, A.P., Stentiford, F.W.: Visual attention for region of interest coding in jpeg 2000. Journal of Visual Communication and Image Representation 14, 232–250 (2003)
4. Liu, T., Sun, J., Zheng, N., Tang, X., Shum, H.: Learning to detect a salient object. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
5. Liu, F., Gleicher, M.: Automatic image retargeting with fisheye-view warping. In: Proc. of ACM Symposium on User Interface Software and Technology, pp. 153–162 (2005)
6. Suh, B., Ling, H., Bederson, B.B., Jacobs, D.W.: Automatic thumbnail cropping and its effectiveness. In: Proc. of ACM Symposium on User Interface Software and Technology, pp. 95–104 (2003)
7. Hou, X., Zhang, L.: Dynamic visual attention: Searching for coding length increments. In: Neural Information Processing Systems, pp. 681–688 (2008)
8. Valenti, R., Sebe, N., Gevers, T.: Image saliency by isocentric curvedness and color. In: IEEE Intl. Conf. on Computer Vision (2009)
9. Bruce, N., Tsotsos, J.: Saliency based on information maximization. In: Neural Information Processing Systems, pp. 155–162 (2005)
10. Wang, Z., Li, B.: A two-stage approach to saliency detection in images. In: IEEE Intl. Conf. on Acoustics, Speech and Signal Processing, pp. 965–968 (2008)
11. Miyazato, K., Kimura, A., Takagi, S., Yamato, J.: Real-time estimation of human visual attention with dynamic bayesian network and mcmc-based particle filter. In: IEEE Intl. Conf. on Multimedia and Expo., pp. 250–257 (2009)
12. Achanta, R., Hemami, S., Estraday, F., Susstrunk, S.: Frequency-tuned salient region detection. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1597–1604 (2009)
13. Hou, X., Zhang, L.: Saliency detection: A spectral residual approach. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
14. Viswanath Gopalakrishnan, Y.H., Rajan, D.: Salient region detection by modeling distributions of color and orientation. IEEE Trans. on Multimedia 11, 892–905 (2009)
15. Han, J., Ngan, K.N., Li, M., Zhang, H.J.: Unsupervised extraction of visual attention objects in color images. IEEE Trans. on Circuits and Systems for Video Technology 16, 141–145 (2006)
16. Ma, Y.F., Zhang, H.J.: Contrast-based image attention analysis by using fuzzy growing. In: Proc. ACM Multimedia, pp. 374–381 (2003)
17. Ko, B.C., Nam, J.Y.: Object-of-interest image segmentation based on human attention and semantic region clustering. Virtual Journal for Biomedical Optics 1, 2462–2470 (2006)
18. Lampert, C.H., Blaschko, M.B., Hofmann, T.: Efficient subwindow search: A branch and bound framework for object localization. IEEE Trans. on Pattern Analysis and Machine Intelligence 31, 2129–2142 (2009)

19. An, S., Peursum, P., Liu, W., Venkatesh, S.: Efficient algorithms for subwindow search in object detection and localization, pp. 264–271 (2009)
20. Lawler, E.L., Wood, D.E.: Branch-and-bound methods: A survey. Operations Research 14, 699–719 (1966)
21. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE Trans. on Pattern Analysis and Machine Intelligence 20, 1254–1259 (1998)
22. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. on System, Man and Cybernetics 9, 62–66 (1979)

Modified Hybrid Bronchoscope Tracking Based on Sequential Monte Carlo Sampler: Dynamic Phantom Validation

Xióngbiāo Luó[1], Tobias Reichl[2], Marco Feuerstein[2], Takayuki Kitasaka[3], and Kensaku Mori[1,4]

[1] Graduate School of Information Science, Nagoya University, Japan
[2] Computer Aided Medical Procedures, Technische Universität München, Germany
[3] Faculty of Information Science, Aichi Institute of Technology, Japan
[4] Information and Communications Headquarters, Nagoya University, Japan

Abstract. This paper presents a new hybrid bronchoscope tracking method that uses an electromagnetic position sensor, a sequential Monte Carlo sampler, and its evaluation on a dynamic motion phantom. Since airway deformation resulting from patient movement, respiratory motion, and coughing can significantly affect the rigid registration between electromagnetic tracking and computed tomography (CT) coordinate systems, a standard hybrid tracking approach that initializes intensity-based image registration with absolute pose data acquired by electromagnetic tracking fails when the initial camera pose is too far from the actual pose. We propose a new solution that combines electromagnetic tracking and a sequential Monte Carlo sampler to address this problem. In our solution, sequential Monte Carlo sampling is introduced to recursively approximate the posterior probability distributions of the bronchoscope camera motion parameters in accordance with the observation model based on electromagnetic tracking. We constructed a dynamic phantom that simulates airway deformation to evaluate our proposed solution. Experimental results demonstrate that the challenging problem of airway deformation can be robustly modeled and effectively addressed with our proposed approach compared to a previous hybrid method, even when the maximum simulated airway deformation reaches 23 mm.

1 Introduction

During minimally invasive diagnosis and surgery of lung and bronchus cancer, bronchoscopy is a useful tool that enables physicians to perform transbronchial biopsies (TBB) to obtain samples of suspicious tumors and to treat or remove precancerous tissue. However, it is still difficult to properly localize the biopsy needle in the region of interest (ROI) to sample tissue inside the airway tree because the TBB procedure is usually guided by conventional bronchoscopy, which only provides 2D information (bronchoscopic video images) and needs to be performed inside the very complex bronchial tree structure. To deal with such limitations, navigated bronchoscopy systems have been developed to help the

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 409–421, 2011.

bronchoscopist by fusing pre-interventional and intra-interventional information such as 3D multi-detector CT image data and real-time bronchoscopic video to provide two fundamental functions: (1) visualization of anatomical structures beyond the bronchial walls and the anatomical names of the currently displayed branches; (2) TBB guidance by showing the planned path of the bronchoscope and localizing the current bronchoscope camera inside the airway tree.

To develop such a bronchoscopic navigation system, the exact pose of the bronchoscope camera must be tracked inside the airway tree for which many techniques have been proposed. Image registration-based methods compare the similarities between real and virtual bronchoscopic images generated from pre-interventional CT data [1,2]. However, such an optimization procedure is constrained heavily by its initialization and bifurcation or fold information to be clearly observed on real bronchoscopic images. Sensor-based electromagnetic tracking (EMT) uses a sensing coil (sensor) attached to the tip of the bronchoscope and localized by an electromagnetic tracking system, such as the commercially available superDimension navigation system [3]. However, such navigation systems suffer from the following bottlenecks: (1) sensitivity to localization problems resulting from patient movement (i.e., airway deformation). An EMT measurement usually provides the position and orientation of the bronchoscope camera relative to a fixed, world coordinate system and hence the current measurement under airway deformation does not correspond exactly to the current bronchoscope camera pose; (2) measurement inaccuracies because of magnetic field distortion caused by ferrous metals or conductive material within or close to the working volume. To address airway deformation, Gergel et al. applied particle filtering to all camera positions and orientations acquired by EMT and projected them to a previously segmented centerline of the bronchial tree [4], so they assume a bronchoscope camera that is always moving along the centerline of the airways; however this is a hard constraint since it is easily violated by a bronchoscopist in the operating room. Otherwise, the measurement inaccuracies of EMT are difficult to correct, unless combined with optical tracking [5,6]. Furthermore, a combination of image- and sensor-based methods for bronchoscope tracking was originally proposed by Mori et al. [7]. Their hybrid method was improved by Soper et al. [8] who integrated electromagnetic tracking, image-based tracking, Kalman filtering, and a respiratory motion compensation method using a surrogate sensor. According to their evaluation of the state-of-the-art methods, the hybrid method is a promising means for bronchoscope tracking and definitely outperforms other methods.

In our paper, we modify hybrid bronchoscope tracking using a sequential Monte Carlo (SMC) sampler to improve tracking performance and to deal with the disadvantages of EMT and the restrictions of image-based methods. Bronchoscope tracking based on Bayesian or motion filtering has already been proposed in [9,10]. However, [9,10] only focused on how to improve the initialization of image registration methods without estimating the rotational part of the bronchoscope camera motion. Our proposed method incorporates electromagnetic tracking and a sequential Monte Carlo sampler to directly estimate the posterior

probability distribution of the current bronchoscope camera motion parameters. This modified method significantly increases the accuracy and the robustness of bronchoscope tracking, as shown in our experimental results.

2 SMC Sampler-Based Bronchoscope Tracking

Our modified hybrid bronchoscope tracking method consists of three stages: (1) during camera and hand-eye calibration, we apply camera calibration to obtain the intrinsic parameters of the bronchoscope camera and employ hand-eye calibration to perform electromagnetic sensor and camera alignment; (2) the CT-to-physical space registration step obtains the initial rigid registration between the EMT and CT coordinate systems. We can use a landmark-based or a landmark-free method to calculate this transformation; (3) the sequential Monte Carlo sampler-based camera motion estimation stage estimates the posterior probability distribution of the current bronchoscope camera motion parameters and determines the estimated camera pose at the maximal probability to correspond to the current bronchoscope camera pose.

Since the first two stages of the proposed method closely resemble the work of Luo et al. [11], we do not describe them here. We focus on modeling and predicting the bronchoscope camera motion based on a sequential Monte Carlo sampler and electromagnetic tracking.

Sequential Monte Carlo samplers such as frameworks [12,13,14] are a generalized class of algorithms dealing with the state estimation problem for nonlinear/non-Gaussian dynamic systems that sequentially sample a set of weighted particles from a sequence of probability distributions defined upon essentially arbitrary spaces using importance sampling and resampling mechanisms. They have been used previously for vision on the basis of structure from motion (SFM), for example, the usage of a general Monte Carlo sampler for SFM in the work of Forsyth et al. [15] and the investigation of particle filtering for simultaneous localization and mapping (SLAM) in [16].

Generally, sequential Monte Carlo samplers are quite similar: samples are determinately drifted and stochastically diffused to approximate the posterior probability distributions of interest. We use an SMC sampler, which resembles the approach of Qian et al. in [17], and only sample the 3-D camera motion parameters; however, Qian et al. sampled the feature correspondences for motion depth determination. We use sequential importance sampling with resampling (SIR) at each iteration to estimate the posterior probability distribution of current bronchoscope camera motion.

2.1 SMC Sampler

Before camera motion estimation, in this section, we briefly review the sequential Monte Carlo sampler based on the SIR scheme.

Suppose a set of state vectors $\mathcal{X}_i = \{\mathbf{x}_i : i = 1, ..., N\}$ and similarly a set of measurements with their history $\mathcal{Y}_i = \{\mathbf{y}_i : i = 1, ..., N\}$, where N is the number

of states or measurements. The sampler using the SIR scheme constructs and approximates the posterior probability distribution $p(\mathbf{x}_i|\mathcal{Y}_i)$ of the current state vector $\mathbf{x}_i$, given all available information, for example, the previous posterior probability distribution $p(\mathbf{x}_{i-1}|\mathcal{Y}_{i-1})$. To estimate $p(\mathbf{x}_i|\mathcal{Y}_i)$, the SIR algorithm first generates a set of random samples $\mathcal{X}_i^k = \{\mathbf{x}_i^k : k = 1, ..., M\}$ with associated weights $\mathcal{W}_i^k = \{w_i^k : k = 1, ..., M\}$ (M is the sample size) at time i based on the previous posterior probability distribution $p(\mathbf{x}_{i-1}|\mathcal{Y}_{i-1})$ and the current measurement $\mathbf{y}_i$. After that, $p(\mathbf{x}_i|\mathcal{Y}_i)$ is approximated by these samples with respect to $\mathbf{x}_i^k$ and w_i^k [13]:

$$p(\mathbf{x}_i|\mathcal{Y}_i) \approx \sum_{k=1}^{M} w_i^k \delta(\mathbf{x}_i - \mathbf{x}_i^k), \tag{1}$$

where $\delta(\cdot)$ is the Dirac delta function. w_i^k can be calculated by

$$w_i^k \propto w_{i-1}^k \frac{p(\mathbf{y}_i|\mathbf{x}_i^k)p(\mathbf{x}_i^k|\mathbf{x}_{i-1}^k)}{q(\mathbf{x}_i^k|\mathbf{x}_{i-1}^k, \mathbf{y}_i)}, \tag{2}$$

where the proposal $q(\cdot)$ is called an importance density function that affects the degree of sample degeneracy. Usually, it is convenient to choose $q(\cdot)$ as the prior: $q(\mathbf{x}_i^k|\mathbf{x}_{i-1}^k, \mathbf{y}_i) = p(\mathbf{x}_i^k|\mathbf{x}_{i-1}^k)$, then $w_i^k \propto w_{i-1}^k p(\mathbf{y}_i|\mathbf{x}_i^k)$ [13].

Basically, a pseudo-code description of an SMC sampler using SIR can be generalized in **Algorithm 1** as follows:

Algorithm 1. SMC Sampler Using SIR Scheme [12]

At $i = 0$, generate M samples $\mathcal{X}_0^k = \{\mathbf{x}_0^k : k = 1, ..., M\}$:
Set initial importance density $q(\mathbf{x}_0^k|\mathbf{x}_0^k, \mathbf{y}_0) = p(\mathbf{x}_0^k)$;
for $k = 1$ **to** M **do**
    Draw sample $\{(\mathbf{x}_0^k, w_0^k)\} \sim q(\mathbf{x}_0^k|\mathbf{x}_0^k, \mathbf{y}_0)$;
    Assign the sample with weights w_0^k;
Compute total weights: $W_0 = \sum_{k=1}^{M} w_0^k$, and normalization: $w_0^k = W_0^{-1} w_0^k$;
for $i = 1$ **to** N **do**
    Calculate the effective sample size: ESS [17], define a threshold: TSS;
    if $ESS < TSS$ **then**
        Resample $\{(\mathbf{x}_{i-1}^k, w_{i-1}^k)\}$ to obtain $\{(\hat{\mathbf{x}}_{i-1}^k, \hat{w}_{i-1}^k)\}$;
    else
        Set $\{(\hat{\mathbf{x}}_{i-1}^k, \hat{w}_{i-1}^k)\} = \{(\mathbf{x}_{i-1}^k, w_{i-1}^k)\}$;
    for $k = 1$ **to** M **do**
        Draw sample $\{(\mathbf{x}_i^k, w_i^k)\} \sim q(\mathbf{x}_i^k|\mathbf{x}_{i-1}^k, \mathbf{y}_i)$;
        Weight $w_i^k \propto \hat{w}_{i-1}^k \omega_i^k$ where incremental importance weight
        ω_i^k is defined as: $\omega_i^k = p(\mathbf{y}_i|\mathbf{x}_i^k)$;
    Compute total weights and normalize each weight: $w_i^k = W_i^{-1} w_i^k$;
    Output current estimated state vector $\tilde{\mathbf{x}}_i = \sum_{k=1}^{M} w_i^k \mathbf{x}_i^k$

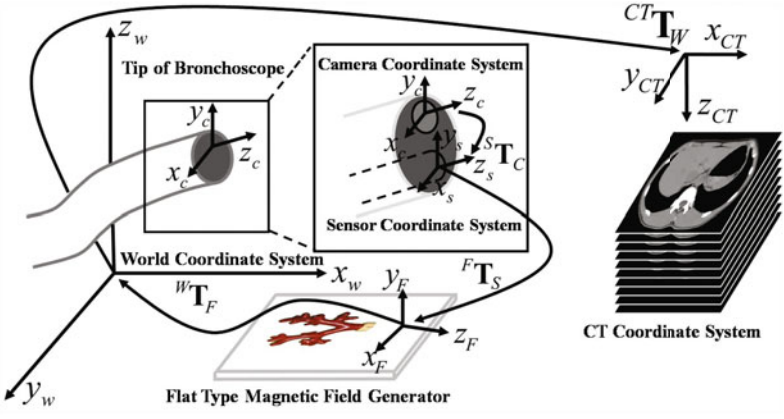


Fig. 1. Relationship between coordinate systems in our navigated bronchoscopy

2.2 Definitions of Bronchoscopic Camera Motion

We must define the coordinate systems to be used since bronchoscope tracking seeks a transformation matrix ${}^{CT}\mathbf{T}_C$ including translation ${}^{CT}\mathbf{t}_C$ and rotation ${}^{CT}\mathbf{R}_C$ from the bronchoscope camera coordinate system to the CT coordinate system. Fig. 1 outlines the relationships and transformation matrices between each coordinate system. ${}^{F}\mathbf{T}_S$ describes the relationship between the sensor and magnetic field coordinate systems. ${}^{W}\mathbf{T}_F$ is from the magnetic field coordinate system to the world coordinate system, and ${}^{CT}\mathbf{T}_W$ is from the world coordinate system to the CT coordinate system. We formulate the relationship between the sensor and world coordinate systems as ${}^{W}\mathbf{T}_S^{(i)} = {}^{W}\mathbf{T}_F {}^{F}\mathbf{T}_S^{(i)}$, where ${}^{F}\mathbf{T}_S^{(i)}$ is the i-th sensor output. Additionally, the transformation between the camera and the sensor (both attached at the bronchoscope tip) is represented by ${}^{S}\mathbf{T}_C$.

In our study, we use the SMC sampler to predict the posterior probability distributions for the bronchoscope camera pose parameters. The camera motion state is described by translation ${}^{CT}\mathbf{t}_C$ and rotation ${}^{CT}\mathbf{R}_C$ from the bronchoscope camera coordinate system to the CT coordinate system. For the rotation part, we use a quaternion but not a rotation matrix ${}^{CT}\mathbf{R}_C$ in our implementation. The quaternion has been demonstrated to be very powerful to characterize the rotation part since it has such advantages as compactness and the avoidance of discontinuous jumps compared to other representations (e.g., Euler angles).

A quaternion representation of rotation can be conveniently considered as a normalized vector with four components:

$$\mathbf{q} = [\, q_0 \quad q_x \quad q_y \quad q_z \,], \qquad {q_0}^2 + {q_x}^2 + {q_y}^2 + {q_z}^2 = 1. \tag{3}$$

Global motion state $\mathbf{x}_i$ that corresponds to the current camera frame can be parameterized by a seven-dimensional vector:

$$\mathbf{x}_i = \left[\begin{array}{cc} {}^{CT}\mathbf{q}_C^{(i)} & {}^{CT}\mathbf{t}_C^{(i)} \end{array} \right], \tag{4}$$

where i means the camera motion state at time i or denotes the i-th electromagnetic tracking result.

According to a sequential Monte Carlo sampler, each random sample $(\mathbf{x}_i^k, w_i^k)$ represents a potential pose of the bronchoscope camera and involves an important weight defined as the similarities between the real and virtual bronchoscopic images in our case. A random sample set $\mathcal{S}_i^k = \{(\mathbf{x}_i^k, w_i^k) : k = 1, 2, 3, ..., M\}$ is used to approximate the posterior probabilistic density of the current bronchocope camera pose at time i.

2.3 SMC Sampler for Camera Motion Estimation

Our proposed hybrid bronchoscope camera motion tracking process is mainly performed by the following steps described in this section.

After parameterizing the current camera motion state $\mathbf{x}_i$ involved with the SMC sampler, bronchoscope tracking continuously estimates the posterior probability distribution $p(\mathbf{x}_i|\mathcal{Y}_i)$ using a set of random samples $\mathcal{S}_i^k$, where the sample weights are proportional to $p(\mathbf{y}_i|\mathbf{x}_i^k)$, as defined in **Algorithm 1**. To obtain these random samples $\mathcal{S}_i^k$, the SMC sampler requires the probabilistic model $p(\mathbf{x}_i^k|\mathbf{x}_{i-1}^k)$ for the state dynamic between the time steps and likelihood function (or an important density function) $q(\mathbf{x}_i^k|\mathbf{x}_{i-1}^k, \mathbf{y}_i)$ for the observations (or measurements) shown in Eq. 2. Additionally, to characterize a random sample $\mathcal{S}_i^k$, the weight w_i^k also needs to be determined by incremental importance weight ω_i^k that equals $p(\mathbf{y}_i|\mathbf{x}_i^k)$. Therefore, the following steps are implemented for the SMC sampler to estimate the bronchoscope camera motion.

[Step 1] State Dynamic. During this state transition step, the bronchoscope motion dynamic at frame i is usually characterized as a second order process that is described by a second order difference equation [17]

$$\mathbf{x}_i^k = U\mathbf{x}_{i-1}^k + V\mathbf{n}_i^k, \tag{5}$$

where the matrix U describes the deterministic drift part of the state dynamic model and depends on the EMT measurements $\mathbf{y}_i$ and $\mathbf{y}_{i-1}$ while the matrix V represents the stochastic diffusion component of the state dynamic model or describes the uncertainty of inter-frame camera motion defined on the basis of Eq. 4. We note that $\mathbf{n}_i^k$ is an independent stochastic variable or a noise term that is discussed in the following paragraph.

Since we have no prior knowledge of the bronchoscope camera movement, we utilize a random walk model to characterize $p(\mathbf{x}_i^k|\mathbf{x}_{i-1}^k)$ for the pointwise state evaluation. As bronchoscopic frames are used as image sources, the changes of the motion parameters are usually quite small. For example, in our case the frame rate of the bronchoscope camera is 30 frames per second; however, the typical moving speed of the camera is around 10 mm per second, so the magnitude of inter-frame motion changes at 0.33 mm per second. Therefore, we used a

random walk on the basis of normal density with respect to noise vector $\mathbf{n}_i^k$: $\mathbf{n}_i^k \sim \mathcal{N}(\mu, \sigma^2)$ to approximate the state dynamic in accordance with Eq. 5 [18]:

$$p(\mathbf{x}_i^k|\mathbf{x}_{i-1}^k) \propto \frac{1}{\sqrt{2\pi\sigma}} \exp(-(V^{-1}(\mathbf{x}_i^k - U\mathbf{x}_{i-1}^k) - \mu)^2/2\sigma^2), \tag{6}$$

After undergoing a random walk based on normal density, the drifted and diffused state $\mathbf{x}_i^k$ has a probabilistic distribution in accordance with Eq. 6.

[Step 2] Observation Model. A good choice of the important density function $q(\mathbf{x}_i^k|\mathbf{x}_{i-1}^k, \mathbf{y}_i)$ can alleviate the sample degeneracy problem. In the SIR algorithm, it is appropriately chosen as prior density $p(\mathbf{x}_i^k|\mathbf{x}_{i-1}^k)$ [13], as mentioned above. We follow this choice: $q(\mathbf{x}_i^k|\mathbf{x}_{i-1}^k, \mathbf{y}_i) = p(\mathbf{x}_i^k|\mathbf{x}_{i-1}^k)$. Therefore, in our case, the observation density $p(\mathbf{y}_i|\mathbf{x}_i)$ can be decided by:

$$p(\mathbf{y}_i|\mathbf{x}_i = \mathbf{x}_i^k) \propto w_i^k (\sum_{j=1}^{M} w_i^j)^{-1}. \tag{7}$$

We clarify that the observation $\mathbf{y}_i$ is defined as the EMT measurement and modeled as $\mathbf{y}_i = H\mathbf{x}_i$, where H is the observation matrix and is usually defined as the transformation from the CT to the EMT coordinates.

[Step 3] Determination of Sample Weight. During the two steps described above, a sample weight w_i^k must be computed to assess the sample performance.

In our study, a sample weight w_i^k is defined as the similarity between the current real bronchoscopic image $\mathbf{I}_R^{(i)}$ and the virtual bronchoscopic image $\mathbf{I}_V$ generated using estimated virtual camera parameters $\mathbf{x}_i^k$ based on a volume rendering technique. Based on the selective image similarity measure [2], after the division of images $\mathbf{I}_R^{(i)}$ and $\mathbf{I}_V$ into subblocks and the selection of subblocks, we use a modified mean squared error (*MoMSE*) to calculate the similarity:

$$MoMSE(\mathbf{I}_R^{(i)}, \mathbf{I}_V) = \frac{1}{|A^{(i)}|} \sum_{D \in A^{(i)}} \frac{1}{|D|} \sum_{D} \left((\mathbf{I}_R^{(i)} - \overline{{}^{D}\mathbf{I}_R^{(i)}}) - (\mathbf{I}_V - \overline{{}^{D}\mathbf{I}_V}) \right)^2, \tag{8}$$

where $|A^{(i)}|$ is the number of selected subblocks in the list of selected subblocks $A^{(i)}$, and $\overline{{}^{D}\mathbf{I}_R^{(i)}}$ and $\overline{\mathbf{V}_D}$ are the respective mean intensities of all subblocks D of $\mathbf{I}_R^{(i)}$ and $\mathbf{I}_V$. The mean intensities of $\mathbf{I}_R^{(i)}$ and $\mathbf{I}_V$ may be different in an actual bronchoscopic image because of the different strengths of the light sources. To reduce this effect, $\overline{{}^{D}\mathbf{I}_R^{(i)}}$ and $\overline{{}^{D}\mathbf{I}_V}$ are subtracted from each pixel.

The weight w_i^k can be formulated as

$$w_i^k = MoMSE(\mathbf{I}_R^{(i)}, \mathbf{I}_V(\mathbf{x}_i^k)). \tag{9}$$

Finally, in our case, the output of the SMC sampler for the current estimated motion state can be determined in accordance with w_i^k:

$$\tilde{\mathbf{x}}_i = \arg\max_{w_i^k} \{(\mathbf{x}_i^k, w_i^k)\}, \tag{10}$$

that is, sample $\tilde{\mathbf{x}}_i$ with maximal weight $\tilde{w}_i$ corresponds to the maximal similarity between the current bronchoscope camera frame and the generated virtual frame.

Our modified hybrid bronchoscope tracking based on an SMC sampler can be summarized in **Algorithm 2** as follows.

Algorithm 2. SMC Sampler-Based Bronchoscope Tracking

input : Bronchoscopic video images $\mathbf{I}_R^{(i)}$, CT-based virtual images $\mathbf{I}_V$, electromagnetic sensor measurements ${}^{W}\mathbf{T}_S^{(i)}$

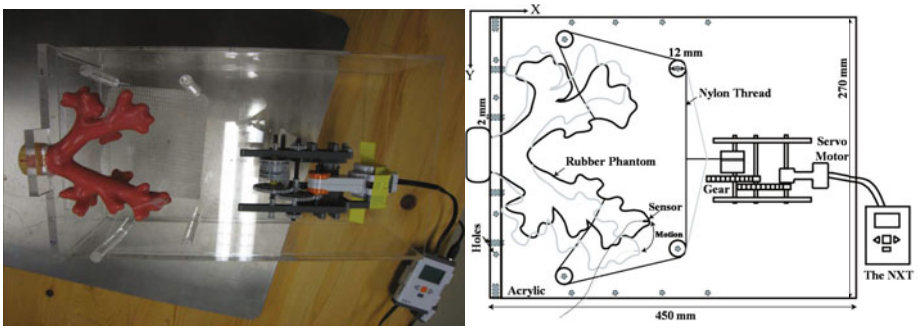
output: A series of estimates ${}^{CT}\widetilde{\mathbf{T}}_C^{(i)}$ of the bronchoscope camera poses

Before SMC sampling:
1. Camera and hand-eye calibration to calculate ${}^{S}\mathbf{T}_C$;
2. CT-to-physical space registration for ${}^{CT}\mathbf{T}_W$;

Start SMC sampling $\Leftrightarrow$ 3. Compute ${}^{CT}\widetilde{\mathbf{T}}_C^{(i)}$
Initialization: At $i = 0$,
Compute ${}^{CT}\mathbf{T}_C^{(0)} = {}^{CT}\mathbf{T}_W {}^{W}\mathbf{T}_S^{(0)} {}^{S}\mathbf{T}_C$, observation: ${}^{CT}\mathbf{T}_C^{(0)} \Leftrightarrow \mathbf{y}_0$;
Generate M samples $\mathcal{X}_0^k = \{\mathbf{x}_0^k : k = 1, ..., M\}$:
for $k = 1$ **to** M **do**
  Draw sample $\{(\mathbf{x}_0^k, w_0^k)\} \sim p(\mathbf{x}_0^k),\ p(\mathbf{x}_0^k) = \frac{1}{M}$;
  $\mathbf{x}_0^k = \mathbf{y}_0$;
  $w_0^k = MoMSE(\mathbf{I}_R^{(0)}, \mathbf{I}_V(\mathbf{x}_0^k))$, according to Eq. 9;
Compute total weights: $W_0 = \sum_{k=1}^{M} w_0^k$, and normalization: $w_0^k = W_0^{-1} w_0^k$;
for $i = 1$ **to** N **do**
  Calculate effective sample size: ESS [17], define a threshold: TSS;
  if $ESS < TSS$ **then**
    Resample $\{(\mathbf{x}_{i-1}^k, w_{i-1}^k)\}$ to obtain $\{(\hat{\mathbf{x}}_{i-1}^k, \hat{w}_{i-1}^k)\}$;
  else
    Set $\{(\hat{\mathbf{x}}_{i-1}^k, \hat{w}_{i-1}^k)\} = \{(\mathbf{x}_{i-1}^k, w_{i-1}^k)\}$;
  Compute ${}^{CT}\mathbf{T}_C^{(i)} = {}^{CT}\mathbf{T}_W {}^{W}\mathbf{T}_S^{(i)} {}^{S}\mathbf{T}_C$, observation: ${}^{CT}\mathbf{T}_C^{(i)} \Leftrightarrow \mathbf{y}_i$;
  for $k = 1$ **to** M **do**
    Draw sample $\{(\mathbf{x}_i^k, w_i^k)\} \sim p(\mathbf{x}_i^k | \mathbf{x}_{i-1}^k)$ by:
    Drift and diffusion: $\mathbf{x}_{i-1}^k \Longrightarrow \mathbf{x}_i^k$ according to **[Step 1]**;
    Calculate observation densities $p(\mathbf{y}_i | \mathbf{x}_i)$ according to **[Step 2]**;
    Weight: $w_i^k = MoMSE(\mathbf{I}_R^{(i)}, \mathbf{I}_V(\mathbf{x}_i^k))$ according to **[Step 3]**;
  Compute total weights: $W_i = \sum_{k=1}^{M} w_i^k$;
  Normalization: $w_i^k = W_i^{-1} w_i^k$;
  The current estimated state $\tilde{\mathbf{x}}_i$: $\tilde{\mathbf{x}}_i = \arg\max_{w_i^k} \{(\mathbf{x}_i^k, w_i^k)\}$;
  Return: $\tilde{\mathbf{x}}_i \Longleftrightarrow {}^{CT}\widetilde{\mathbf{T}}_C^{(i)}$

3 Experimental Results

For evaluating the performance of our proposed tracking method, we manufactured a dynamic bronchial phantom (Fig. 2) to simulate breathing motion. We connected the rubber phantom to a motor using nylon threads. A LEGO



(a) (b)

Fig. 2. Dynamic motion phantom: (a) picture of real phantom and (b) drawing of phantom movement

Mindstorm (LEGO, Denmark) was utilized as power source to generate movement. With the controller part (NXT: a programmable robotics kit included in LEGO Mindstorm), we can manipulate the motor motion including the directions and the rotational speeds. The phantom simulates respiratory motion when the thread changes its length. We can adjust the amount of simulated motion, and its maximum deformation is about 24 mm.

For dynamic phantom validation, we compare four tracking schemes: (a) Solomon et al. [3], only using EMT, (b) Mori et al. [7], intensity-based image registration directly initialized by the EMT results, (c) Luo et al [11], the better one of two proposed schemes in [11], and (d) our method, as described in Section 2.3.

Table 1 shows the quantitative results of the evaluation of the methods. Here we counted the number of frames that were successfully registered by visually inspecting the similarities between the real and virtual images. The maximum

Table 1. Comparison of registered results (the unit of maximal motion is mm)

Experi.	Maximal	Number (percentage) of successfully registered frames			
(frames)	motion	Solomon et al. [3]	Mori et al. [7]	Luo et al. [11]	Our method
A(1285)	6.13	850 (66.1%)	958 (74.6%)	1034 (80.5%)	1224 (95.3%)
B(1326)	11.82	783 (59.0%)	863 (65.1%)	1018 (76.8%)	1244 (93.8%)
C(1573)	18.75	894 (56.8%)	972 (61.8%)	1153 (73.3%)	1431 (91.0%)
D(1468)	23.61	716 (48.8%)	850 (57.9%)	1036 (70.6%)	1300 (88.6%)
Total(5652)		**3243 (57.4%)**	**3643 (64.5%)**	**4241 (75.0%)**	**5199 (92.0%)**

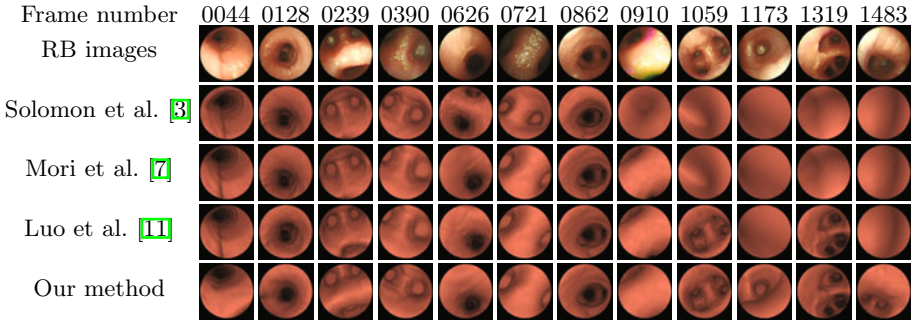
simulated respiratory motion for different experiments is also shown in Table 1. Our proposed method significantly improved the tracking performance. Furthermore, examples of experiments C and D for successfully registered frames are displayed in Fig. 3, which shows examples of real bronchoscopic (RB) images and corresponding virtual bronchoscopic (VB) images generated from the camera parameters predicted by each method.

4 Discussion

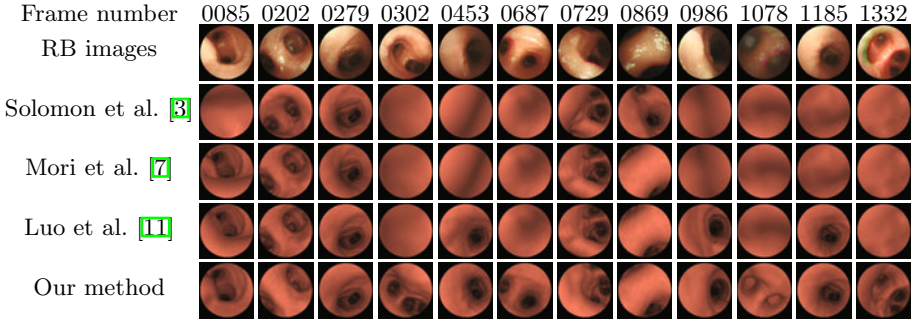
The objective of this study is to design and improve the performance of hybrid bronchoscope tracking under airway deformation during bronchoscopic navigation, in particular, to deal with the limitations of electromagnetic tracking. We used a sequential Monte Carlo sampler to modify previous hybrid bronchoscope tracking methods. According to the experimental results, the posterior probability distributions of the bronchocope camera poses are almost completely approximated using the sequential Monte Carlo sampler. Hence we improved our previous proposed hybrid tracking methods [7,11] in various aspects.

As for the previous hybrid method [7], its tracking robustness and accuracy usually suffer from the following: (1) dependencies on the initialization of image registration and visible characteristic structures (i.e., folds or bifurcations of the bronchi) for similarity computation; (2) airway deformation, in particular respiratory motion. For the registration step (an optimization procedure), the optimizer is unavoidably trapped in local minima. We have already addressed these limitations and improved the tracking performance by modifying the initialization of image registration in our previous work [11]. In this study, our modified method was more effectively disengaged from these constraints using a sequential Monte Carlo sampler, compared to our previous methods [7,11]. We greatly approximate the posterior densities of the state parameters by collecting a set of random samples and sequentially predict the camera motion parameters on the basis of the importance sampling, which provides the ability to maintain potential importance modes that either they are confirmed or moved to be the subsequent observations. This results in our proposed method that can avoid the optimization registration algorithm which is trapped in local minima in most cases and particularly has the ability to automatically retrieve the tracking loss even in case of image artifacts. Hence, our method shows the best tracking performance in Table 1 and Fig. 3, compared to the previous methods.

However, in our experiments, the modified methods still failed to correctly register all RB and VB frames when continuously tracking the bronchoscope for the following reasons: (1) the dynamic error of EMT (because of the ferrous material contained inside the bronchoscope), as mentioned in Section 1, affected the observation accuracy; (2) our simulated breathing motion is rather big and not realistic enough. Currently it is only in the left-right and superior-inferior directions for the peripheral lung. The trachea does not move. The magnitude of the motion can be adjusted to $6 \sim 24$ mm. However, for a real patient, respiratory motion is greatest in the superior-inferior direction ($\sim$ 9 mm), moderate in



(a) Examples of experiment C



(b) Examples of experiment D

Fig. 3. Results of bronchoscope tracking for different methods under simulated breathing motion using our dynamic phantom. The top row shows selected frame numbers and the second row shows their corresponding phantom RB images. The other rows display virtual bronchoscopic images generated from tracking results using the methods of Solomon et al. [3], Mori et al. [7], Luo et al. [11], and our method. Our proposed method shows the best performance.

the anterior-posterior direction ($\sim$ 5 mm), and lowest in the left-right direction ($\sim$ 1 mm) [19].

Additionally, the average runtime of our proposed method per frame (1.7 seconds) is higher than that of the previous hybrid method (0.5 seconds), because each random sample must compute its weight based on the similarities between real and virtual images; this is really time-consuming.

5 Conclusions and Future Work

This paper presented a modified hybrid bronchoscope tracking method that used an electromagnetic position sensor and a sequential Monte Carlo sampler and evaluation on a dynamic phantom. We used a sequential Monte Carlo sampler to approximate the posterior probability distributions of the bronchoscope camera motion parameters. Experimental results demonstrated that the modified

method gives impressive approximations to the bronchoscope camera motion and successfully registered a total of 5199 (92.0%) bronchoscopic images, increasing the tracking performance by 17.0% compared to the state-of-the-art hybrid method. We conclude that our method significantly alleviates the sensitivity to the localization problems of electromagnetic tracking that usually result from airway deformation, particularly respiratory motion. Our future work includes experiments on patient datasets using our proposed method in the operating room and improvement of its computational efficiency.

References

1. Bricault, I., et al.: Registration of real and CT-derived virtual bronchoscopic images to assist transbronchial biopsy. IEEE TMI 17, 703–714 (1998)
2. Deguchi, D., et al.: Selective image similarity measure for bronchoscope tracking based on image registration. MedIA 13, 621–633 (2009)
3. Solomon, S.B., et al.: Three-dimensionsal CT-guided bronchoscopy with a real-time electromagnetic position sensor: a comparison of two image registration methods. Chest 118, 1783–1787 (2000)
4. Gergel, I., et al.: Particle filtering for respiratory motion compensation during navigated bronchoscopy. In: Proceedings of SPIE, vol. 7625 (2010) 76250W
5. Deguchi, D., et al.: A method for bronchoscope tracking by combining a position sensor and image registration. Proceedings of CARS 1281, 630–635 (2005)
6. Feuerstein, M., et al.: Magneto-optical tracking of flexible laparoscopic ultrasound: Model-based online detection and correction of magnetic tracking errors. IEEE TMI 28, 951–967 (2009)
7. Mori, K., Deguchi, D., Akiyama, K., Kitasaka, T., Maurer Jr., C.R., Suenaga, Y., Takabatake, H., Mori, M., Natori, H.: Hybrid Bronchoscope Tracking Using a Magnetic Tracking Sensor and Image Registration. In: Duncan, J.S., Gerig, G. (eds.) MICCAI 2005. LNCS, vol. 3750, pp. 543–550. Springer, Heidelberg (2005)
8. Soper, T.D., et al.: In vivo validation of a hybrid tracking system for navigation of an ultrathin bronchoscope within peripheral airways. IEEE TBME 57, 736–745 (2010)
9. Nagao, J., Mori, K., Enjouji, T., Deguchi, D., Kitasaka, T., Suenaga, Y., Hasegawa, J.-i., Toriwaki, J.-i., Takabatake, H., Natori, H.: Fast and Accurate Bronchoscope Tracking Using Image Registration and Motion Prediction. In: Barillot, C., Haynor, D.R., Hellier, P. (eds.) MICCAI 2004. LNCS, vol. 3217, pp. 551–558. Springer, Heidelberg (2004)
10. Deligianni, F., Chung, A., Zhong, G.: Predictive Camera Tracking for Bronchoscope Simulation with CONDensation. In: Duncan, J.S., Gerig, G. (eds.) MICCAI 2005. LNCS, vol. 3749, pp. 910–916. Springer, Heidelberg (2005)
11. Luo, X., et al.: Towards hybrid bronchoscope tracking under respiratory motion: evaluation on a dynamic motion phantom. In: Proceedings of SPIE, vol. 7625 (2010) 76251B
12. Liu, J.S., et al.: Sequential Monte Carlo methods for dynamic systems. Journal of the American Statistical Association 93, 1032–1044 (1998)
13. Arulampalam, M., et al.: A tutorial on particle filters for nonlinear/non-gaussian Bayesian tracking. IEEE TSP 50, 174–188 (2002)
14. Moral, P.D., et al.: Sequential Monte Carlo samplers. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 68, 411–436 (2006)

15. Forsyth, D., et al.: The joy of sampling. IJCV 41, 109–134 (2001)
16. Pupilli, M.: Particle filtering for real-time camera localisation. PhD thesis, University of Bristol, UK (2006)
17. Qian, G., et al.: Structure from motion using sequential Monte Carlo methods. IJCV 59, 5–31 (2004)
18. Doucet, A., et al.: On sequential monte carlo sampling methods for Bayesian filtering. Statistics and Computing 10, 197–208 (2000)
19. Soper, T.D., et al.: A model of respiratory airway motion for real-time tracking of an ultrathin bronchoscope. In: Proceedings of SPIE, vol. 6511 (2007) 65110M

Affine Warp Propagation for Fast Simultaneous Modelling and Tracking of Articulated Objects

Arnaud Declercq and Justus Piater

Montefiore Institute, University of Liège, Belgium
Arnaud.Declercq@ulg.ac.be, Justus.Piater@ulg.ac.be

Abstract. We propose a new framework that allows simultaneous modelling and tracking of articulated objects in real time. We introduce a non-probabilistic graphical model and a new type of message that propagates explicit motion information for realignment of feature constellations across frames. These messages are weighted according to the rigidity of the relations between the source and destination features. We also present a method for learning these weights as well as the spatial relations between connected feature points, automatically identifying deformable and rigid object parts. Our method is extremely fast and allows simultaneous learning and tracking of nonrigid models containing hundreds of feature points with negligible computational overhead.

1 Introduction

Articulated object models have become an active research topic in recent years. In the vast majority of applications, an object is represented by a graphical model connecting rigid body parts [1,2]. While those models are usually designed by hand, some solutions have been proposed to automatically learn articulated models from tracks of feature points [3,4]. Unfortunately, robust feature tracking proves to be a challenge on its own in long sequences.

Tracking those points becomes much easier when they can rely on a feature graph to assist them. Since good tracking requires a model and a model is learnt based on good tracking, the most obvious solution is to simultaneously track and learn the feature graph. While some solutions have already been proposed for offline learning [5] or learning based on a short initialisation period [6], very few are dedicated to online learning. This domain is indeed very challenging: not only must both learning and tracking be computed in real time, but also tracking must rely on an incomplete intermediate model. Although we addressed the latter issue with an uncertain Gaussian model that explicitly accounts for its predictive power [7], we were then only able to achieve real-time tracking on very small feature graphs. Most of the computational time was not dedicated to learning but to tracking using the popular Belief Propagation solution [8,9] to propagate position likelihoods between nodes of the graph.

Here, we propose a new tracking solution that sacrifices the multimodality of nonparametric, probabilistic methods for a significant gain in computational

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 422–435, 2011.

efficiency. The main idea here is to keep the benefits of a propagation scheme to share information between nodes, but formulated for a new, non-probabilistic feature graph. While each node of the graph still represents the current position of its associated feature point, propagated messages do not convey a potential function but simply the information needed to align feature points in the new image. Thanks to this solution, we will show that large feature graphs can be simultaneously learnt and tracked in real time. As feature points, we will present our method using edgels (i.e. points along edges) because they are more robust to illumination changes and lack of texture. This also allows us to show that, even if each edgel only provides a 2D translational constraint, it can be tracked properly in a 6-dimensional affine space thanks to our propagation method.

After discussing some background in Section 2, we will introduce in Section 3 the image alignment of a template based on the motion of a complete set of feature points. This will give us a first idea of the information that has to be conveyed in order to align features in a new image. In Section 4, we will consider each feature as a template but with a connection only to its direct neighbours, and show how information from further features can be propagated in order to be used in its alignment. In Section 5, we explain how spatial relations between feature points are learnt and used during Affine Warp Propagation. Finally, experimental results are presented in Section 6.

2 Related Work

Various methods have been proposed for simultaneous learning and tracking of rigid graph models [7,10,11]. The main drawback of these methods is their limitation to rigid objects or so small that feature displacements can be assumed spatially coherent.

Unsupervised learning of articulated models from a video sequence usually relies on existing feature trajectories that are processed off-line [3,4]. Ramanan et al. [5] proposed an off-line unsupervised method that simultaneously discovers, tracks and learns articulated models of animals from video. Unfortunately, their method is slow and requires the whole video to be treated as a block, making it impossible to adapt to an on-line process. Krahnstoever et al. [6] presented an automatic on-line acquisition and initialisation of articulated models. Their method extracts independently moving surfaces and tracks them using Expectation-Maximisation during a short initialisation period. An articulated model is deduced from these motions and is used for tracking in subsequent images. Since no model update is provided, it is strongly dependent on the key-frames selected for learning. Finally, Droin et al. [12] developed a method to incrementally segment the rigid parts of an object on-line. It maintains a set of possible models learnt from previous frames, and uses them for tracking. Although interesting, their technique suffers two main drawbacks: it requires a foreground extraction, and it doesn't learn or use any spatial relations during tracking.

In terms of feature-graph tracking, the most popular solution is Belief Propagation (BP) [8,9] and its derivatives like Nonparametric Belief Propagation or

Sequential Belief Propagation [9,13,14] that combine BP with particle filters. Although our method is completely different from BP since it is a non-probabilistic solution, its message passing process is strongly inspired from it. Apart from that, our method is more closely related to image alignment techniques such as the Lucas-Kanade algorithm [15,16].

3 Image Alignment via Affine Warps

Consider, first, that we have a model of the object we want to track. This model consists of a weighted set of n edgels and their positions $X = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ in the model's referential. In order to align this model with a set of target positions $T = (\mathbf{t}_1, \ldots, \mathbf{t}_n)$ in a given image, we use an affine warp:

$$W(\mathbf{x}_i; \mathbf{p}) = \begin{bmatrix} 1+p_1 & p_3 & p_5 \\ p_2 & 1+p_4 & p_6 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{1}$$

where $\mathbf{p} = [p_1, p_2, p_3, p_4, p_5, p_6]$ corresponds to the warp's parameters, and $\mathbf{x}_i = [x_i, y_i]$. Alignment of the model with target positions is given by the warp that minimises the sum of squared residuals

$$\mathbf{R} = \sum_i w_i \|W(\mathbf{x}_i; \mathbf{p}) - \mathbf{t}_i\|^2 \tag{2}$$

with w_i representing the weight of point i.

In the case of tracking, we can assume that a current estimate of $\mathbf{p}$ is known and target positions $T = (\mathbf{t}_1, \ldots, \mathbf{t}_n)$ can be obtained using the new image (using the closest contour points to each edgel for example). Alignment of the template in the new image is then obtained by minimising $\mathbf{R}$ for the incremental warp $W(\mathbf{x}, \Delta\mathbf{p})$:

$$\mathbf{R} = \sum_i w_i \|W(W(\mathbf{x}_i; \mathbf{p}); \Delta\mathbf{p}) - \mathbf{t}_i\|^2 \tag{3}$$

Then parameters $\mathbf{p}$ are updated such that

$$W(\mathbf{x}; \mathbf{p}_{\text{new}}) = W(W(\mathbf{x}; \mathbf{p}_{\text{old}}); \Delta\mathbf{p}). \tag{4}$$

In order to solve the warp update, the expression in equation 3 is linearised by performing a first-order Taylor expansion on $W(W(\mathbf{x}; \mathbf{p}); \Delta\mathbf{p})$ to give:

$$\mathbf{R} = \sum_i w_i \left\| W(W(\mathbf{x}_i; \mathbf{p}); \mathbf{0}) + \frac{\partial W}{\partial \mathbf{p}} \Delta\mathbf{p}_i - \mathbf{t}_i \right\|^2 \tag{5}$$

Since $W(\mathbf{x}; \mathbf{0})$ is the identity warp, $W(W(\mathbf{x}; \mathbf{p}); \mathbf{0})$ then simplifies to $W(\mathbf{x}; \mathbf{p})$. Following the notational convention that partial derivatives with respect to a

column vector are laid out as a row vector and with $W(\mathbf{x};\mathbf{p}) = [W_x, W_y]^T$, the Jacobian $\frac{\partial W}{\partial \mathbf{p}}$ of the warp at equation 5 is given by:

$$\frac{\partial W}{\partial \mathbf{p}} = \begin{bmatrix} \frac{\partial W_x}{\partial p_1} & \frac{\partial W_x}{\partial p_2} & \cdots & \frac{\partial W_x}{\partial p_6} \\ \frac{\partial W_y}{\partial p_1} & \frac{\partial W_y}{\partial p_2} & \cdots & \frac{\partial W_y}{\partial p_6} \end{bmatrix} = \begin{bmatrix} x & 0 & y & 0 & 1 & 0 \\ 0 & x & 0 & y & 0 & 1 \end{bmatrix} \tag{6}$$

Notice that $\frac{\partial W}{\partial \mathbf{p}}$ is computed for $W(W(\mathbf{x};\mathbf{p});\Delta\mathbf{p})$, which means that values used in equation 6 are the current coordinates of the points in the image.

The solution to the minimisation of equation 5 is obtained by setting to zero its partial derivatives with respect to $\Delta\mathbf{p}$:

$$\Delta\mathbf{p} = \mathbf{H}^{-1} \sum_i w_i \left[\frac{\partial W}{\partial \mathbf{p}}\right]^T D_i \tag{7}$$

where $D_i = [\mathbf{t}_i - W(\mathbf{x}_i;\mathbf{p})]^T$ is the displacement required of point i, and $\mathbf{H}$ is the $n \times n$ Hessian matrix (here with $n = 6$):

$$\mathbf{H} = \sum_i w_i \left[\frac{\partial W}{\partial \mathbf{p}}\right]^T \left[\frac{\partial W}{\partial \mathbf{p}}\right] \tag{8}$$

The warp update then consists of iteratively applying equations 7 and 4 until estimates of the parameters $\mathbf{p}$ converge (in the case of affine transformation, one iteration is sufficient since the system is linear in the parameters which wouldn't be the case with a parameter such as orientation for example).

The solution obtained in equations 7 and 8 is interesting because it means that $\Delta\mathbf{p}$ is computed using only the two matrices $\mathbf{H}$ and $\mathbf{S} = \sum_i w_i \left[\frac{\partial W}{\partial \mathbf{p}}\right]^T D_i$ whose sizes are independent of the number of points used. The fact that both matrices are computed as a sum over the points is also advantageous since information provided by new points will be easily added to the current matrices. Those two conditions met, a message containing them seems an attractive candidate to propagate motion information.

4 Affine Warp Propagation

In the case of a feature graph, no feature point is connected to a global model. Each one has only access to a set of learnt relations with its direct neighbours. Since the motion of an edgel is locally ambiguous, it will require information from the biggest possible neighbourhood. Moreover, if relations between edgels have to be learnt, it would be more convenient to express edgel configurations in an affine space instead of simply 2D space. In order to compute the affine warp needed to align a point and its surrounding in a new image, information will then have to be propagated in the graph. Before getting into the details of which information is required and how to propagate it, let's first consider the representation of the non-probabilistic graphical model we wish to learn.

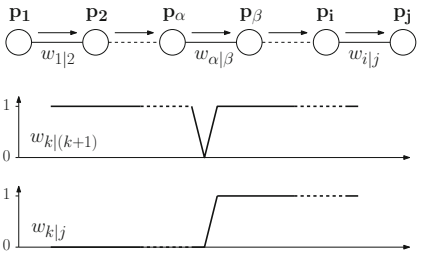


Fig. 1. Simple line graph situation where all spatial relations between connected points are rigid except for the relation between α and β. This means that this graph represents two uncorrelated rigid sets of points: one on the left and one on the right of the relation between α and β. The learnt weights between a node k and the node to its right are given by $w_{k|(k+1)}$ (centre panel). In the case where messages are propagated from left to right, the bottom panel illustrates the resulting influence $w_{k|j}$ on the rightmost point j of each upstream point k. Here, the null weight $w_{\alpha|\beta}$ eliminates the influence on p_j of the points that are not in the same rigid set.

4.1 Non-probabilistic Graphical Model Definition

A node i of a graph contains the set of parameters $\mathbf{p}_i$ used to warp the corresponding edgel i and its neighbourhood from the origin to a position that aligns them with the current image. In that sense, a node is similar to the template discussed in Section 3 except that it doesn't have direct connection to all the points used for its alignment.

An edge going from node i to node j represents two types of information. First, it contains the learnt affine parameters $\mathbf{r}_{i|j}$ of i as they were previously observed in the affine space of j, i.e. the parameters that align i and its surroundings from the origin to their observed positions in the space of j. Secondly, it contains the weight $w_{i|j}$ node j should give to the information coming from node i. This weight is directly related to the rigidity of the relation: the lower the correlation between two features, the lower the weight and then the smaller the influence of messages passing through this connection. This means that information coming from points behind a non-rigid connection will have no influence on image alignment (see Figure 1 for an example).

4.2 Message Definition

In this section, we will use, for the sake of explanation, the simple line graph shown in Figure 1. If node j had access to the displacement D_k of all the points on the graph, its warp update using equation 7 would be

$$\Delta\mathbf{p}_j = \mathbf{H}_j^{-1} \sum_{k \leq j} w_{k|j} \left[\frac{\partial W_{k|j}}{\partial \mathbf{p}_j} \right]^T D_k, \tag{9}$$

where we use $\frac{\partial W_{k|j}}{\partial \mathbf{p}_j}$ to indicate that the Jacobian is computed for $W(W(\check{\mathbf{r}}_{k|j}; \mathbf{p}_j); \mathbf{0})$, i.e. for the projection in the image coordinates of the 2D position $\check{\mathbf{r}}_{k|j} = [r_{x,k|j}, r_{y,k|j}]$

of node k in the referential of node j. Now, if we define $w_{k|j} = w_{k|i}w_{i|j}$ for the graph of Figure 1, the sum can be decomposed the following way:

$$\mathbf{S}_j = \sum_{k \leq j} w_{k|j} \left[\frac{\partial W_{k|j}}{\partial \mathbf{p}_j}\right]^T D_k \tag{10}$$

$$= w_j \left[\frac{\partial W_{j|j}}{\partial \mathbf{p}_j}\right]^T D_j + w_{i|j} \sum_{k \leq i} w_{k|i} \left[\frac{\partial W_{k|i}}{\partial \mathbf{p}_i}\right]^T D_k \tag{11}$$

$$= S_j + w_{i|j}\mathbf{S}_i \tag{12}$$

where $w_j = w_{j|j}$ is the weight node j gives to its feature point and, more importantly, where we made the assumption that

$$\frac{\partial W_{k|j}}{\partial \mathbf{p}_j} = \frac{\partial W_{k|i}}{\partial \mathbf{p}_i} \tag{13}$$

This assumption means we consider that point k is projected in the same image coordinates by nodes i and j, i.e. $W(\check{\mathbf{r}}_{k|j}; \mathbf{p}_j) = W(\check{\mathbf{r}}_{k|i}; \mathbf{p}_j)$ which is actually correct if nodes i and j are linked by a perfectly rigid spatial relation. Since the weights are null for non-rigid relations, this assumption seems valid. Unfortunately, we will see in Section 4.3 that even if the weights are all correct (which is not guaranteed during the learning phase), small numerical errors can trigger a drift from the correct tracking result. While this will motivate a more general formulation in Section 4.3, we propose to continue with this assumption for now in order to understand more easily what type of information a message should contain.

Decomposing $\mathbf{H}_j$ in a similar way to Equation 12, we obtain:

$$\mathbf{H}_j = H_j + w_{i|j}\mathbf{H}_i \tag{14}$$

Equations 12 and 14 mean that the warp update for a node j based on all the points of the graph in Figure 1 can be computed using only the information from itself and that accumulated by node i. In the case of affine warps, S_j and H_j are given by:

$$S_j = w_j \begin{bmatrix} x_j D_{x,j} & x_j D_{y,j} & y_j D_{x,j} & y_j D_{y,j} & D_{x,j} & D_{y,j} \end{bmatrix}^T \tag{15}$$

$$H_j = w_j \begin{bmatrix} x_j^2 & 0 & x_j y_j & 0 & x_j & 0 \\ 0 & x_j^2 & 0 & x_j y_j & 0 & x_j \\ x_j y_j & 0 & y_j^2 & 0 & y_j & 0 \\ 0 & x_j y_j & 0 & y_j^2 & 0 & y_j \\ x_j & 0 & y_j & 0 & 1 & 0 \\ 0 & x_j & 0 & y_j & 0 & 1 \end{bmatrix} \tag{16}$$

where we used $D_j = [D_{x,j}, D_{y,j}]$, $x_j = p_{x,j}$, $y_j = p_{y,j}$. Note that H_j has a lot of null or identical elements. Without any loss of information, we can thus reduce it to the vector:

$$\overline{H}_j = w_j \begin{bmatrix} 1 & x_j & y_j & x_j^2 & x_j y_j & y_j^2 \end{bmatrix}^T \tag{17}$$

A message containing the two vectors $\mathbf{S}$ and $\overline{\mathbf{H}}$ is then enough to convey all the information needed to align the nodes in the new image. Notice that the messages are not expressed in the same space as the feature points or the nodes. This way, displacements can be accumulated through small 12-element messages in order to compute the affine alignment of the nodes without any loss of information. Moreover, an affine warp can be computed (given that enough nodes have been visited) with those messages while each node only needs to provide a translation. With a propagation scheme inspired from Belief Propagation [8,9], the computation of the warp update for each node i can be summarised in 3 steps :

1. Initialise the information for each node k of the graph using equations 15 and 17 and send a first message to each neighbouring node $i \in \mathcal{N}(k)$:

$$\mathbf{S}_{ki}^{0} = S_k; \tag{18}$$

$$\overline{\mathbf{H}}_{ki}^{0} = \overline{H}_k; \tag{19}$$

2. Propagate the information between the nodes for l iterations (for a message sent from node i to node j):

$$\mathbf{S}_{ij}^{l} = S_i + \sum_{k \in \mathcal{N}(i) \setminus j} w_{k|i} \mathbf{S}_{ki}^{l-1} \tag{20}$$

$$\overline{\mathbf{H}}_{ij}^{l} = \overline{H}_i + \sum_{k \in \mathcal{N}(i) \setminus j} w_{k|i} \overline{\mathbf{H}}_{ki}^{l-1} \tag{21}$$

3. Compute the update of the warp parameters for each node j:

$$\mathbf{S}_j = S_j + \sum_{k \in \mathcal{N}(j)} w_{k|j} \mathbf{S}_{kj}^{l} \tag{22}$$

$$\overline{\mathbf{H}}_j = \overline{H}_j + \sum_{k \in \mathcal{N}(j)} w_{k|j} \overline{\mathbf{H}}_{kj}^{l} \tag{23}$$

$$\mathbf{H}_j \leftarrow \overline{\mathbf{H}}_j \tag{24}$$

$$\Delta \mathbf{p}_j = \mathbf{H}_j^{-1} \mathbf{S}_j \tag{25}$$

with $\mathcal{N}(j)$ representing the set of neighbouring nodes to node j. This solution provides a very fast propagation method that allows each node to align itself in a new image using as much information as possible provided by other feature points. Notice that no information is lost in the message passing process. This means that updating the warp parameters by Equation 7 using a template or by Equation 25 using the message passing process will give exactly the same result (given that the messages went once through all the nodes of the template).

4.3 Message Correction

In the previous section we made the assumption that $W(\check{\mathbf{r}}_{k|j}; \mathbf{p}_j) = W(\check{\mathbf{r}}_{k|i}; \mathbf{p}_j)$ in order to obtain Equations 12 and 14. This assumption means that a node k

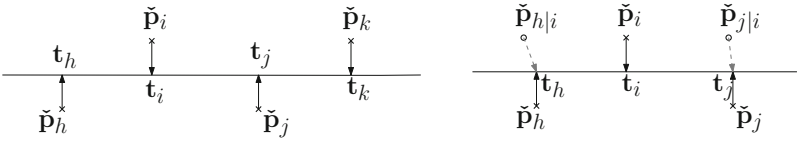


Fig. 2. Consider the edgels in Figure 2a. Those edgels should be in straight line but are not, due to tracking inaccuracy. If nodes i and j are aligned in this new image using displacement of their direct neighbours, they will drift even further from each other while they should align along the contour. On the other hand, if they know where their neighbours should be and compute their displacement from there, the nodes will be able to correct for the current drift (see Figure 2b where $\check{\mathbf{p}}_{h|i} = W(\check{\mathbf{r}}_{h|i}; \mathbf{p}_i)$ and $\check{\mathbf{p}}_{j|i} = W(\check{\mathbf{r}}_{j|i}; \mathbf{p}_i)$ represent the positions nodes h and j should have with respect to node i).

should be expected in the same position by all the nodes belonging to the same rigid block. Even if all the nodes are indeed rigidly connected to each other, this assumption might not be correct simply because of numerical inaccuracies in the tracking result. Consider, for example, the case of tracking a set of nodes as shown in Figure 2. Edgels have been extracted along a line segment and tracked for a few frames. Due to some inaccuracy in the tracking, the edgels are not in a straight line anymore. If node i aligns itself using its two direct neighbours' motion information, it will be pushed up while it should actually go down (it will also be dramatically scaled down on the vertical direction in case of affine nodes). Similarly, node j will be forced to move down making it drift even further from the correct tracking result. The reason for this problem is quite simple: each node acts as the model described in Section 3 but does not itself evaluate the displacement of the points used. This displacement is indeed provided by each individual node without any consideration of whether it belongs to the model of another node or not. If the assumption $W(\check{\mathbf{r}}_{k|j}; \mathbf{p}_j) = W(\check{\mathbf{r}}_{k|i}; \mathbf{p}_i)$ is verified, the points used by a model are located exactly where they are supposed to be, and the displacement information is therefore correct. If it is not verified (for numerical reason for example), the node will simply try to match this new configuration of edgels to the current image instead of trying to get back to its initial configuration. For the example of Figure 2a, this means matching the v-shape form by h, i and j to a line segment.

By learning the correct relative positions and using them as the origin of the displacement (as shown in Figure 2b), the proper relative position of the nodes can be maintained and the drift problem eliminated. Concerning a message to a node i, this means that every occurrence of a position $\check{\mathbf{p}}_k = W(\check{\mathbf{r}}_{k|k} = \mathbf{0}; \mathbf{p}_k)$ must be shifted to the expected position $\check{\mathbf{p}}_{k|i} = W(\check{\mathbf{r}}_{k|i}; \mathbf{p}_i)$. By the same idea, every displacement $D_k = \mathbf{t}_k - \check{\mathbf{p}}_k$ must be replaced with the expected displacement $D_{k|i} = \mathbf{t}_k - \check{\mathbf{p}}_{k|i}$. Notice that the target $\mathbf{t}_k$ is not modified and is then only an approximation of the true target $\check{\mathbf{p}}_{k|i}$ should have provided. Indeed, the correct target $\mathbf{t}_{k|i}$ cannot be computed since the information about $\check{\mathbf{p}}_{k|i}$ is merged into $\mathbf{S}_i$

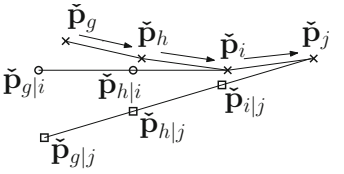


Fig. 3. Position correction: $\check{\mathbf{p}}_k$ represents the current 2D position of the edgel associated with node k, $\check{\mathbf{p}}_{k|i}$ its position as expected by i and $\check{\mathbf{p}}_{k|j}$ its position as expected by j. In this example, nodes should be align along a straight line while they are more in a curve configuration. If node i has already corrected the position of the nodes on its left into a straight line, all is left to do for node j is to apply a single warp to all the $\check{\mathbf{p}}_{k|i}$ to align them with the $\check{\mathbf{p}}_{k|j}$. The warp needed to do that is the one that aligns the affine parameters $\mathbf{p}_i$ to $\mathbf{p}_{i|j}$.

and $\overline{\mathbf{H}}_i$. However, since the drift is corrected at each frame, it is kept very small and $\mathbf{t}_k$ is therefore a good estimate of $\mathbf{t}_{k|i}$. With these modifications applied, we can see in Figure 2b that node i will receive coherent information, causing it to move downwards as needed.

The correction of the positions and displacements of all the points used in the alignment of a node j is a little tricky because a node j has only access to the information ($\mathbf{r}_{i|j}$and $\mathbf{p}_i$) related to its direct neighbours and the messages $m^l_{j,i} = \{\mathbf{S}^l_{ji}, \overline{\mathbf{H}}^l_{ji}\}$ they send. This means that a message coming from a neighbour i must already be corrected for i (since no spatial relation has been learnt with further points) and then adapted for j. Figure 3 shows an example of this situation where, again, we consider the case where all the points should be in a straight line while they are obviously not. So, assume that all the positions $\check{\mathbf{p}}_k$ and displacements D_k of the points k included in the message $m^l_{j,i}$ have already been corrected into $\check{\mathbf{p}}_{k|i}$ and $D_{k|i}$ respectively. In order to obtain the positions $\check{\mathbf{p}}_{k|j}$ expected by j, the only thing left to do is to adapt the positions $\{\check{\mathbf{p}}_{1|i}, \ldots, \check{\mathbf{p}}_{i|i}\}$ proposed by node i to the positions $\{\check{\mathbf{p}}_{1|j}, \ldots, \check{\mathbf{p}}_{i|j}\}$ expected by node j. The correction is the same for all the positions included in the message. Given that $\mathbf{p}_i$ and $\mathbf{p}_{i|j}$ are known by node j they can be used to compute the transformation needed to go from $\check{\mathbf{p}}_{k|i}$ to $\check{\mathbf{p}}_{k|j}$. This transformation is simply given by

$$\mathbf{x}_{k|j} = W(W(\check{\mathbf{p}}_{k|i}; \mathbf{p}_i^{-1}); \mathbf{p}_{i|j}) \tag{26}$$

for any $\check{\mathbf{p}}_{k|i}$. Equation 26 means that a position $\check{\mathbf{p}}_{k|i}$ is warped back into the referential of node i and then warped into the image using the warp parameters $\mathbf{p}_{i|j}$.

Since we do not have direct access to positions $\check{\mathbf{p}}_{k|i}$, we will have to apply this transformation directly to the message, i.e. to $\mathbf{S}^l_{ji}$ and $\overline{\mathbf{H}}^l_{ji}$. Using the notation

$$\begin{bmatrix} x_{k|j} \\ y_{k|j} \\ 1 \end{bmatrix} = W(W(\check{\mathbf{p}}_{k|i}; \mathbf{p}_i^{-1}); \mathbf{p}_{i|j}) = \begin{bmatrix} a & c & v \\ b & d & w \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k|i} \\ y_{k|i} \\ 1 \end{bmatrix} \tag{27}$$

for the correction of the points in the message, we will now apply this correction directly to the Hessian part of the message corrected by node i and sent to j, i.e.

$$\overline{\mathbf{H}}^{l}_{ji|i} = \sum_{k} w_{k|i} \begin{bmatrix} 1 & x_{k|i} & y_{k|i} & x^2_{k|i} & x_{k|i}y_{k|i} & y^2_{k|i} \end{bmatrix}^T \tag{28}$$

The corrected Hessian part $\overline{\mathbf{H}}^{l}_{ji|j}$ is obtained using equation 27 on each of its terms of, which gives

$$\overline{\mathbf{H}}^{l}_{ji|j} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ v & a & c & 0 & 0 & 0 \\ w & b & d & 0 & 0 & 0 \\ v^2 & 2av & 2cv & a^2 & 2ac & c^2 \\ vw & aw+bv & cw+dv & ab & ad+bc & cd \\ w^2 & 2bw & 2dw & b^2 & 2bd & d^2 \end{bmatrix} \overline{\mathbf{H}}^{l}_{ji|i} \tag{29}$$

The correction of $\mathbf{S}_{ij}$ is somewhat more difficult because it also depends on $D_{k|i} = \mathbf{t}_k - \check{\mathbf{p}}_{k|i}$. The target position $\mathbf{t}_k$ is not modified by the correction, so we replace $D_{k|i} = [D_{x,k|i}, D_{y,k|i}]$ by $[t_{x,k} - x_{k|i}, t_{y,k} - y_{k|i}]$ in

$$\mathbf{S}^{l}_{ij|i} = \sum_{k} w_{k|i} \left[x_{k|i}D_{x,k|i} \; x_{k|i}D_{y,k|i} \; y_{k|i}D_{x,k|i} \;\; y_{k|i}D_{y,k|i} \; D_{x,k|i} \; D_{y,k|i} \right]^T \tag{30}$$

and apply the same correction as for $\overline{\mathbf{H}}^{l}_{ji|i}$ to yield

$$\mathbf{S}^{l}_{ij|j} = \begin{bmatrix} a & 0 & c & 0 & v & 0 \\ 0 & a & 0 & c & 0 & v \\ b & 0 & d & 0 & w & 0 \\ 0 & b & 0 & d & 0 & w \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{S}^{l}_{ij|i} + \begin{bmatrix} 0 & v & 0 & a & c & 0 \\ 0 & 0 & v & 0 & a & c \\ 0 & w & 0 & b & d & 0 \\ 0 & 0 & w & 0 & b & d \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \overline{\mathbf{H}}^{l}_{ji|i} - \begin{bmatrix} \overline{\mathbf{H}}^{l}_{ji|j}(4) \\ \overline{\mathbf{H}}^{l}_{ji|j}(5) \\ \overline{\mathbf{H}}^{l}_{ji|j}(5) \\ \overline{\mathbf{H}}^{l}_{ji|j}(6) \\ \overline{\mathbf{H}}^{l}_{ji|j}(2) \\ \overline{\mathbf{H}}^{l}_{ji|j}(3) \end{bmatrix} \tag{31}$$

Using these two equations to correct the messages allows us to solve the drift problem by maintaining the nodes at their learnt relative positions, making the tracking more robust to occlusions and clutter.

5 Learning Spatial Relations and Weights

As we noted above, the learnt relations are key to successful tracking with Warp Propagation. Not only do they define the shape of the correlated neighbourhood used in the tracking through their weight, but they also model the expected relative configurations that keep the feature points in proper relative positions. The relative expected configuration $\mathbf{r}_{i|j}$ of a node i with respect to a node j is learnt from the observed affine parameters of i in the affine space of j. Since

those relations are learnt online during tracking, they should be reliable from the first frame even with an obviously incomplete data set to be learnt from. This means that, while the relations are expected to assist in tracking, they cannot exert an overly strong bias that would hamper it. Earlier we proposed an Uncertain Potential Function that solves this problem for visual feature graph tracking with Sequential Belief Propagation by combining an informative Gaussian model learnt from the previous observations with a non-informative part [7]. This potential function for the relative position of a node i expressed in the affine space of a node j was given by

$$\psi_{i|j}(p_i, p_j) = \lambda_{i|j} \mathrm{e}^{-\frac{1}{2}(\mathbf{s}_{i|j}-\mathbf{r}_{i|j})\tilde{\Sigma}_{i|j}^{-1}(\mathbf{s}_{i|j}-\mathbf{r}_{i|j})} + (1-\lambda_{i|j}) \tag{32}$$

where $\mathbf{s}_{i|j}$ represents the observed parameters p_i expressed in the space defined by p_j, $\mathbf{r}_{i|j}$ is the learnt relative configuration (i.e. the mean of the relative configurations already observed), $\lambda_{i|j}$ is the probability that the model is indeed Gaussian and $\tilde{\Sigma}_{i|j}$ is a covariance matrix that accounts for the uncertainty related to the incomplete data set. Due to lack of space here, we refer the interested reader to our earlier work [7] for more details.

While this spatial relation representation is specially designed to be used for the tracking during its learning phase, it is also specific to probabilistic feature graphs, which our representation is not. Nevertheless, very few changes are needed to adapt this uncertain model to our problem. Indeed, the computed mean $\mathbf{r}_{i|j}$ already represents the most likely relative affine configuration we used in Section 4, so we simply have to compute our relational weight using the covariance matrix $\tilde{\Sigma}_{ij}$ and model probability $\lambda_{i|j}$ from the uncertain model [7]:

$$w_{i|j} = \lambda_{i|j} \prod_n \mathrm{e}^{-\frac{\tilde{\Sigma}_{nn,i|j}}{\sigma_{n,i|j}^2}} \tag{33}$$

where $\tilde{\Sigma}_{nn,i|j}$ represents the nth diagonal element of $\tilde{\Sigma}_{i|j}$, and $\sigma_{i|j} = [\sigma_{1,i|j}, \ldots, \sigma_{6,i|j}]$ defines the level of variance accepted for each parameter. This way, the more variance we observe in the relative configuration of two feature points, the less weight each one will give to a message coming from the other.

6 Experiments

In this section, we demonstrate the performance of our method on a set of representative examples of simultaneous learning and tracking. Since we are interested in articulated objects, we propose to use points extracted along their skeleton (let's call them skedgels) in addition to edgels. Those points do not have an appearance in the image and instead rely on edgels to infer their displacement. Models are initialised as shown in the first row of Figure 4 where relations are created between each pair of skedgels within a distance lower than

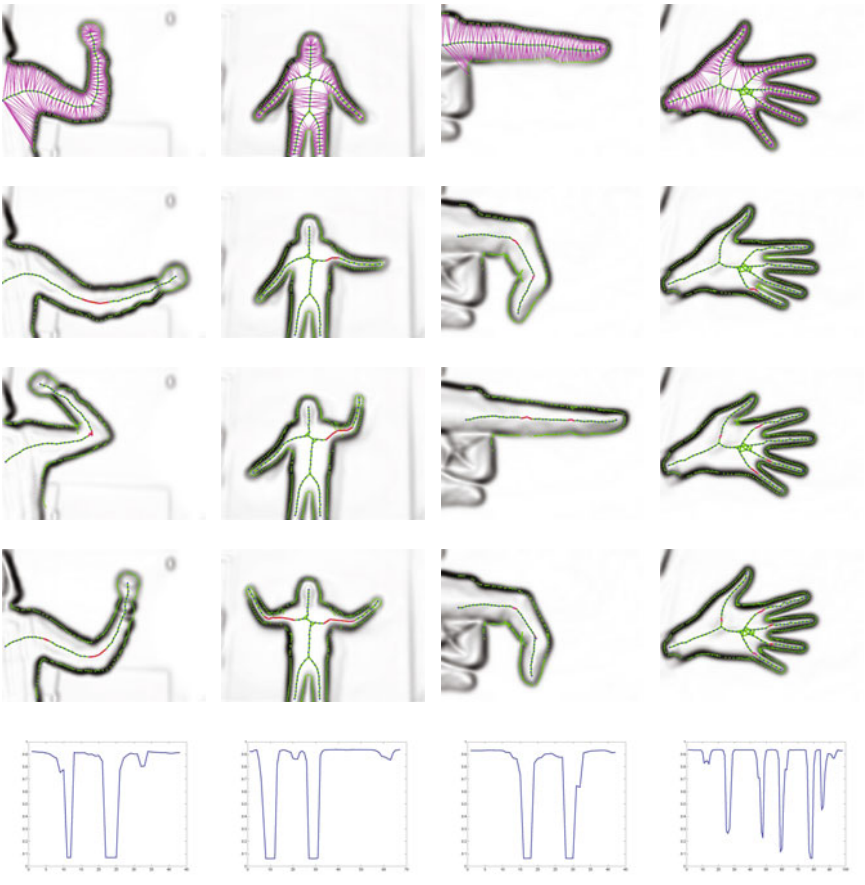


Fig. 4. Examples of simultaneous modelling and tracking of articulated objects. The first row shows the graphs as they are initialised in the first frame. The three central rows show intermediate results during the video (arm: frames 40, 90 and 255, body: 60, 130 and 330, finger: 90, 230, 310 and hand: 70, 230 and 310). Connections in red between skedgels correspond to relations with a weight lower than 0.5. The last row shows the weights of the relations in the last frame of the video. The y-axis corresponds to the weights (range between 0 and 1) and the x-axis correspond to the index of the relation. Although the correspondences of these indices to the feature graphs are not shown, it is evident that there are clean cuts in the graph with weights close to 0 that correspond to non-rigid parts, while the rigid relations have a weight close to one.

a given threshold and between skedgels and edgels using Delaunay triangulation. The model is tracked using Warp Propagation with 10 iterations, and relations between skedgels are updated at each frame with the method explained in Section 5. Results can be found in Figure 4 where the last row represents the relation weights between skedgels at the end of the video. Notice that, while edgels and skedgels on their own would slide long the objects, here their

correct position is maintained thanks to Warp Propagation. Thanks to the learnt relations, the influence neighbourhood is limited to the rigid parts.

On a Pentium Core 2 Duo 2x2 GHz with 2Gb of RAM, the tracking time for each frame (including the likelihood propagation) is between 1.6 and 3.6ms, and the learning time is between 0.3 and 0.7ms. The slowest sequence is the hand with 99 skedgels, 319 edgels and 99 relations. The fastest is the finger with 42 skedgels, 203 edgels and 41 relations.

7 Discussion

We presented a new framework for efficient propagation of alignment information through a feature-point graph. Instead of propagating potential functions as is usually done, we propagate only the motion information needed to align feature points and their surroundings in the image. We showed that this solution allows us to simultaneously track and learn unknown, articulated objects in a few milliseconds per frame, making our solution practical in real-time scenarios even with a large number of feature points. This article focused mainly on tracking but, in the future, it would be interesting to provide a learning scheme for articulated relations instead of simple Gaussian models.

Acknowledgement. This work is supported by a grant from the Belgian National Fund for Research in Industry and Agriculture (FRIA) to A. Declercq and by the EU Cognitive Systems project PACO-PLUS (IST-FP6-IP-027657).

References

1. Sudderth, E.B., Mandel, M.I., Freeman, W.T., Willsky, A.S.: Visual hand tracking using nonparametric belief propagation. In: CVPRW 2004, Washington, DC, USA, vol. 12. IEEE Computer Society, Los Alamitos (2004)
2. Wu, Y., Hua, G., Yu, T.: Tracking articulated body by dynamic markov network. In: ICCV 2003, Washington, DC, USA, p. 1094. IEEE Computer Society, Los Alamitos (2003)
3. Ross, D.A., Tarlow, D., Zemel, R.S.: Unsupervised learning of skeletons from motion. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 560–573. Springer, Heidelberg (2008)
4. Yan, J., Pollefeys, M.: Automatic kinematic chain building from feature trajectories of articulated objects. In: CVPR 2006, Washington, DC, USA, pp. 712–719. IEEE Computer Society, Los Alamitos (2006)
5. Ramanan, D., Forsyth, D.A., Barnard, K.: Building models of animals from video. In: PAMI 2006, vol. 28, pp. 1319–1334 (2006)
6. Krahnstoever, N., Yeasin, M., Sharma, R.: Automatic acquisition and initialization of articulated models. Mach. Vision Appl. 14, 218–228 (2003)
7. Declercq, A., Piater, J.H.: On-line simultaneous learning and tracking of visual feature graphs. In: Online Learning for Classification Workshop, CVPRW 2007 (2007)
8. Yedidia, J.S., Freeman, W.T., Weiss, Y.: Understanding belief propagation and its generalizations, pp. 239–269 (2003)

9. Sudderth, E.B., Ihler, A.T., Freeman, W.T., Willsky, A.S.: Nonparametric belief propagation. In: CVPR 2003, vol. 1, pp. I–605–I–612 (2003)
10. Leordeanu, M., Collins, R.: Unsupervised learning of object features from video sequences. In: CVPR 2005, vol. 1, pp. 1142–1149. IEEE Computer Society, Los Alamitos (2005)
11. Yin, Z., Collins, R.: On-the-fly object modeling while tracking. In: CVPR 2007, pp. 1–8. IEEE Computer Society, Los Alamitos (2007)
12. Drouin, S., Hébert, P., Parizeau, M.: Incremental discovery of object parts in video sequences. Comput. Vis. Image Underst. 110, 60–74 (2008)
13. Briers, M., Doucet, A., Singh, S.S.: Sequential auxiliary particle belief propagation. In: 8th Int. Conf. on. Information Fusion, 2005, vol. 1, p. 8 (2005)
14. Hua, G., Wu, Y.: Multi-scale visual tracking by sequential belief propagation. In: CVPR 2004, vol. 1, pp. 826–833 (2004)
15. Baker, S., Matthews, I.: Lucas-kanade 20 years on: A unifying framework. Int. J. Comput. Vision 56, 221–255 (2004)
16. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: IJCAI 1981, pp. 674–679 (1981)

kPose: A New Representation For Action Recognition

Zhuoli Zhou[1], Mingli Song[1,*], Luming Zhang[1], Dacheng Tao[2], Jiajun Bu[1], and Chun Chen[1]

[1] College of Computer Science, Zhejiang University, Hangzhou, China
[2] School of Computer Engineering, Nanyang Technological University, Singapore

Abstract. Human action recognition is an important problem in computer vision. Most existing techniques use all the video frames for action representation, which leads to high computational cost. Different from these techniques, we present a novel action recognition approach by describing the action with a few frames of representative poses, namely *kPose*. Firstly, a set of pose templates corresponding to different pose classes are learned based on a newly proposed Pose-Weighted Distribution Model (PWDM). Then, a local set of *kPose*s describing an action are extracted by clustering the poses belonging to the action. Thirdly, a further *kPose* selection is carried out to remove the redundant poses among the different local sets, which leads to a global set of *kPose*s with the least redundancy. Finally, a sequence of *kPose*s is obtained to describe the action by searching the nearest *kPose* in the global set. And the proposed action classification is carried out by comparing the obtained pose sequence with each local set of *kPose*. The experimental results validate the proposed method by remarkable recognition accuracy.

1 Introduction

Human action recognition in videos has great potentials in applications such as video surveillance, content-based video search, human-computer interaction, etc. In general, an action recognition process can be divided into three steps briefly: feature extraction, action representation and classification. To extract features over complex conditions like different person appearances, backgrounds, viewpoints and resolutions and keep the representation good enough to carry out robust classification, most conventional approaches employ all the video frames of an action as the representation [1,2,3], which leads to extreme spatial cost for feature storage.

When the action representation is ready, human action recognition becomes a classification problem. There are two groups of action classification methods: time-dependent models and time-independent models. The time-dependent model consists of states linked together wherein each states summarizes the action performance at a certain moment, e.g., Hidden Markov Model [4].

* Corresponding author, brooksong@cs.zju.edu.cn

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 436–447, 2011.

The time-dependent model describes the complicated actions well but lacks discriminative ability for some related action such as *walking* and *jogging*. In contrast, the time-independent models deal with this problem well by providing more discriminative ability, but need extremely high time and memory cost for computation, such as k-Nearest Neighbor (NN) [3].

It is proved that very few frames are enough to perform action recognition [5,6,7,8,9,10], which leads to extraordinarily decreasing in spatial and temporal cost. Schindler et al. [9] combine both shape and flow responses as features and compare 10 different frame lengths from 1 to 10 as action descriptor. Thurau and Hlavác [10] use Histogram of Gradient (HoG) [11] as the basis feature and focus on human pose in each frame by background subtraction and non-negative matrix factorization. Hatun et al. [5] model an action templates as a string of poses which are identified by HoG. Novel actions are matched to templates by applying well-known string comparing method, about a half of poses are needed only. However, these methods use the fixed same number of frames for all actions without considering the different characteristics of each action. On one hand, the frames for an action may be not enough to describe the discriminative information which leads to misclassification; on the other hand, the frames used are more than requires which causes extra computation and storage cost.

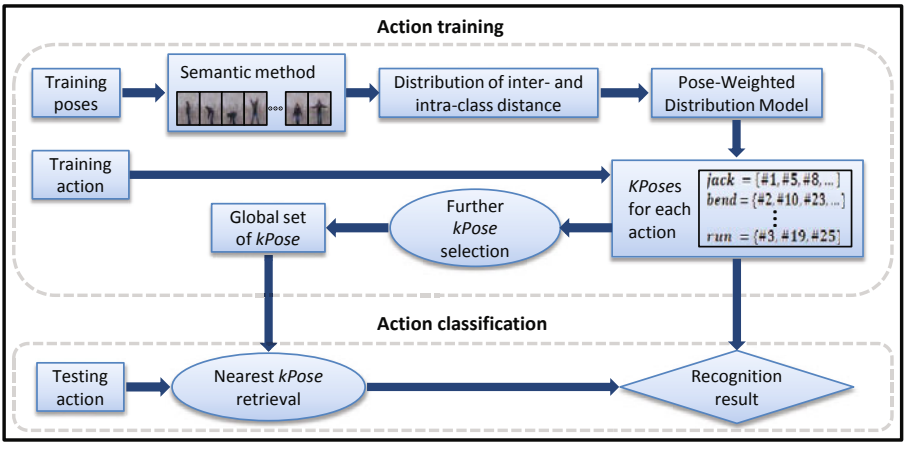


Fig. 1. Overview of our approach

Motivated by these issues, we introduce a novel representation of human action and the corresponding classifying method in this paper. Fig. 1 shows the workflow diagram of the proposed approach. The representative pose, namely *kPose* which is extracted from a class of poses in our approach, is introduced to describe the human action and reflects the main state of the action. Different actions are described by different numbers of *kPose*s where the numbers are decided automatically by pose-weighted distribution model.

In our approach, HoG is extracted as features from each video frame. All the poses belonging to an action are clustered ten times, into a certain number of

classes where the class number is from 1 to 10. To compare the clustering results of different class number, we computed the distances between poses and divided them into two sets: intra- and inter-class distance set. These sets are used to construct the PWDM, which takes the pose templates as directrix and estimates the impact of class number on clustering results. With maximal estimate value, the class number is selected as the *kPose* number for the action. Then, *kPose*s are extracted from each class and forms a local set. However, as it is probable that the local sets from different actions have overlap, further selection is carried out to remove the redundant *kPose*s to obtain a global set for action representation.

For the action classification in our approach, each pose in the video is compared with the *kPose*s in the global set and represented by the the nearest one. So each action can be described by a set of *kPose*s and encoded by a series of numbers indicating the *kPose*s. Finally, a straight forward matching process is carried out to identify the action type by referring to the training actions.

2 Feature for Pose Description

Reliable feature extraction and pose detection are crucial for successful pose-based action recognition. Most difficulties in poses matches arise from background and pose articulation. Local features, such as interest-point, speedup robust feature (SURF) [12], are extracted reliably and robust to different background and localization. Laptev and Lindeberg [13] use Harris corner descriptor [14] as the interest-point in 3D.space-time model where the spatial and temporal neighborhood undergo a translation in time. In a similar fashion, Dollar et al. [15] apply Gabor filter on the spatial and temporal dimensions individually after interest-point extraction. In these approaches, the pose representation is organized as a collection of local features which often leads to the loss of spatial correlation and the decreasing of pose detection rate.

Global features, such as silhouette, edge and optical flow which are obtained through background subtraction or tracking, are powerful because most of the information in the feature are useful for the description of pose articulation. However, the global features are usually sensitive to noise, viewpoints and appearance. To overcome the limitation of global features, Danafar and Gheissari [16] divide the extracted optical-flow into cells, each of which is computed locally. Davis et al. [2] organize a sequence of silhouettes as history energy image (HEI). However, overwriting HEI leads to less discriminative ability for action recognition in moving scene.

Very recently, HoG descriptor shows its robustness in pedestrian detection and recent works [5,10,17] show its power for pose detection in action recognition. Therefore, as a basis feature for describing poses we use the HoG descriptor which divides the oriented gradient of each pixel into cells and modelled to histogram.

3 Pose Weighted Distribution Model

At the beginning of this section, we would like to describe the pose template labelling method which intended to decrease the disagreement between human

labelling and assumed ground-truth which has been discussed in [18]. Then the distribution function of intra- and inter-class distance, i.e., the basement of PWDM, are computed on pose templates. At last, we introduce the PWDM model which estimates the clustering result to decide the class number for each action.

Table 1. Semantic labelling method. Each $1 \sim 2$ bits indicate a articulation of a part of human body, and different value of these bits indicate clear different pose articulations. This method describes 14 pose templates and corresponding class of poses as presented in Fig. 2.

Bit index	Index parts	Value	Meanings
1	Orientation	0	Standing up
		1	Standing up sideways
2-3	The upper part of body	00	0° rotating
		01	45° rotating
		10	90° rotating
4-5	Arm lifted angle	00	0° lifting
		01	90° lifting
		10	180° lifting
6	Arm number	0	One arm
		1	Two arms
7	Legs	0	Not separating
		1	Separating
8	Leg curling	0	Not curling
		1	Curling

It is obviously difficult to manually assign the label to poses because some poses cannot be distinguished even by human. To tackle this problem, we follow a part-based semantic way. Namely, each part of human body corresponds to $1 \sim 2$ bits; any clearly different pose is assigned a value as template. The total of 8 bits binary number, indexes 4 parts of human body and 256 different poses shown in Table 1.

By setting *standing up sideways* as the first one in the set of pose templates, a new pose is added as a novel template once its deviation from all pose templates which have been in the set is detected. After traversing entire data set, 14 of 256 poses are enough to summarize all the poses, which is shown in Fig. 2.

3.1 Intra- and Inter-class Distance

In this part, we describe the distribution of intra- and inter-class distance set based on the set of pose templates.

The pose training set X consists of 14 class of poses which are labelled referring to pose templates. The Euclidean distance of each pose pair (x_i, x_j), no matter which class these poses belong to, is computed as $D(x_i, x_j) = \|x_i - x_j\|$.

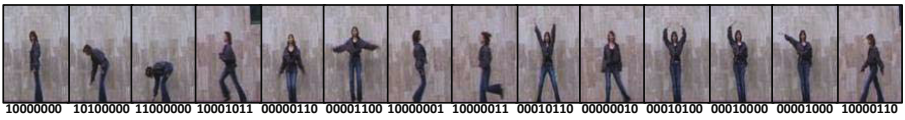


Fig. 2. 14 pose templates and corresponding semantic number. 500 poses selected from actions are labelled manually to these templates. We use *k-means* method to cluster these 500 poses to 14 classes, yet 57 poses are different from our manual label which is understood as vision error.

The intra-class distance is defined as a set S_w (w for within): for all the pose pair (x_i, x_j), if x_i, x_j belong to the same class, $D(x_i, x_j)$ is in S_w; else the $D(x_i, x_j)$ is in the inter-class distance set S_b (b for between).

The rationale behind the proposed method is based on the following assumption: an optimal action clustering solution with k classes can be obtained by traversing all possible class number which ensures the optimal balance between intra- and inter-class distance. This assumption seems very natural: as we increase the class number of clustering, the elements, which are relatively larger than others in intra-class distance set S_w, are moved into inter-class distance set S_b. Finally, the increasing trend of S_b and decreasing trend of S_w are balanced at a certain class number.

We base the balancing procedure on pose templates which are considered as the perfect clustering result, even though there are vision errors. Assuming the distance under Gaussian distribution with identical expected values and variances, the S_w and S_b are described as distribution functions separately:

$$f_w(D) = a_w \cdot e^{-\frac{(D-\mu_w)^2}{2{\sigma_w}^2}} \tag{1}$$

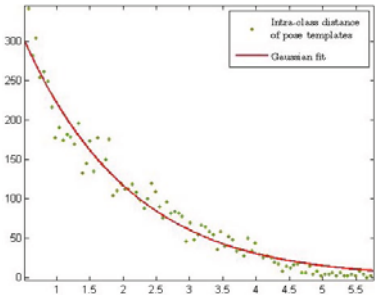
$$f_b(D) = a_b \cdot e^{-\frac{(D-\mu_b)^2}{2{\sigma_b}^2}} \tag{2}$$

As illustrated in Fig. 3. These two distribution functions are used to indicate the weights of different distances. The weighted average value of intra- and inter-class distance over X are computed as: $\mu'_w = \frac{\sum f_w(d_i)d_i}{\sum f_w(d_i)}$ and $\mu'_b = \frac{\sum f_b(d_j)d_j}{\sum f_b(d_j)}$ where $d_i \in S_w, d_j \in S_b$. In the next section, we will discuss how to use these functions and values during the balancing procedure .

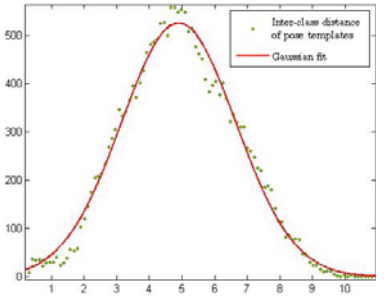
The assumptions above are validated by experiment, where the optimal class number has always been found within 10 time tries (class #1 $\sim$ 10).

3.2 Pose-Weighted Distribution Model Learning

Given a novel action X_i, which consists of several classes of poses, we use f_w and f_b to compute the weight of distance for all pose pairs. Then, we present a method to estimate the class number of the action by comparing the intra- and inter-class distance to the pose templates. However, it is difficult to compare the elements between action X_i and pose templates one by one. Thus the weighted



(a)The distribution of intra-class distance of pose templates



(b)The distribution of inter-class distance of pose templates

Fig. 3. Assuming the distance under Gaussian distribution. $\mu_w = -31.58$, $\sigma_w = 9.86$, $\mu_b = 4,893$ and $\sigma_b = 2.471$ where 500 poses (about 10%) are randomly selected from the data set.

intra-class distance d_w and weighted inter-class distance d_b are computed for each class. For j^{th} class $X_{i,j}$, the computation result is presented by a 3-tuple:

$$R_{i,j} = (n_j, d_b, d_w) \tag{3}$$

where n_j is the class size, d_w and d_b are defined as:

$$d_w = \frac{\sum_{l=1}^{n_j} \sum_{k=l}^{n_j} D(x_l, x_k) \cdot f_w(D(x_l, x_k))}{\sum_{l=1}^{n_j} \sum_{k=l}^{n_j} f_w(D(x_l, x_k))} \tag{4}$$

$$d_b = \frac{\sum_{x_k \notin X_{i,j}} \sum_{l=1}^{n_j} D(x_l, x_k) \cdot f_b(D(x_l, x_k))}{\sum_{x_l \notin X_{i,j}} \sum_{l=1}^{n_j} f_b(D(x_l, x_k))} \tag{5}$$

Then we present a probability density function $P(R_{i,j}|R_0)$, which quantizes the comparability between a class of poses in the new action and the existing pose templates.

$$e(i,j) = P(R_{i,j}|R_0) = P(d_b|\mu'_b)P(d_w|\mu'_w)P(n_j|C_i/k) \tag{6}$$

$e(i,j)$ is the computed probability, where $R_0 = (C_i/k, \mu'_b, \mu'_w), k$ is the class number of the action, and $C_i = \sum_{j=1}^{k} n_j$ is the action size. The three terms in (6) defined as follows:

$$P(d_b|\mu'_b) = (1-\beta_b)^{|d_b - \mu'_b|} \tag{7}$$

$$P(d_w|\mu'_w) = (1-\beta_w)^{|d_w - \mu'_w|} \tag{8}$$

$$P(n_j|C_i/k) = \begin{cases} (1-\beta_c)^{|\frac{n_j - C_i/k}{C_i/k}|} & |\frac{n_j - C_i/k}{C_i/k}| < \varepsilon \\ (1-\beta_c)^{\varepsilon} & \text{others} \end{cases} \tag{9}$$

$P(d_b|\mu'_b)$ is the probability density of intra-class distance, and $P(d_w|\mu'_w)$ is that of inter-class distance. Variance σ_b, σ_w are computed by Gaussian fitting in Section 3.1, and $\frac{\beta_b}{\sigma_b} = \frac{\beta_w}{\sigma_w} = \gamma$. The optimal factor γ has been determined empirically, and has been proved to be stable across different pose selection method for pose templates. $P(n_j|C_i/k)$ constrains the class size to be neither too large nor too smaller.

3.3 *KPose* Based Action Representation

Given an action training set $X = \{X_1, \cdots, X_N\}$, where X_i is the i^{th} action of X, and $X_i = \{x_{i,1}, \cdots, x_{i,C_i}\}$, we define that the minimal *kPose* number is 1 and maximal *kPose* number is 10 which are experimentally enough to find optimal class number. To collect *kPose*s for action X_i, the clustering operation is performed 10 times where the class number varies from 1 to 10 for action X_i.

Table 2. The algorithm of *kPose* calculation and selection

Input:	$X = \{X_1, X_2, \cdots, X_N\}$	//action training set
	f_w, f_b, μ'_w, μ'_b	//*kPose* model parameter
	η	//predefined threshold
Output:	$Y = \{y_1, y_2, \cdots, y_n\}$	//global set of *kPose*s
	$Y_i = \{y'_1, y'_2, \cdots, y'_{n_i}\}$	//*kPose*s for action X_i
1.	**begin**	
2.	*for* $i = 1$ to N do begin	//**Part 1:** *kPose* calculation:
3.	*for* $n = 1$ to 10 do begin	
4.	cluster action X_i to n classes under *global k-means method*	
5.	calculate $E(X_i, n)$	
6.	end	//*for on line 3*
7.	$n_i = \arg\max_n E(X_i, n)$	
8.	calculate Y_i for action X_i	
9.	$Y \leftarrow Y_i \bigcup Y$	
10.	*end while* 1	//**Part 2**: *kPose* global selection:
11.	calculate $r(y_i, y_j)$ for each *kPose* pair (y_i, y_j)	
12.	$r \leftarrow max\ r(y_i, y_j)$	
13.	*if* $(r > \eta)$ do begin	
14.	remove y_j from Y	
15.	replace y_j with y_i for all Y_k	
16.	*else break*	
17.	*end*	//*while on line 12*
18.	**end**	

As we have defined the probability in (6) for a class of poses, a weighted estimate value for the action is defined as below:

$$E(X_i, n) = \frac{\sum_{j=1}^{n} size(j) \cdot e(i, j)}{\sum_{j=1}^{n} size(j)} \tag{10}$$

where $size(j)$ is the size of j^{th} class, and $e(i,j)$ is the probability value of j^{th} class as defined in (6). To make the clustering robust to the clustering kernel and the starting condition, we employ *global k-means* algorithm [19] in our approach, which functions in an incremental way, i.e., before solving the clustering problem ψ with M sets, all intermediate problems with $1, 2, \cdots, M-1$ sets are sequentially solved. This algorithm ensures that the clustering result does not depend on any initial parameter value. Then we find class number corresponding to maximal estimate value $n_{max}(X_i) = \arg\max_{n=1}^{10} E(X_i, n)$ where the number also corresponds to the number of *kPose*s.

For a well clustered action X_i, $X_{i,j} = \{x_{1'}, \cdots, x_{n'_j}\}$ is j^{th} class, and the *kPose* computed as:

$$y(X_{i,j}) = \arg\min \sum_{x_{i,k'}, x_{i,l'} \in X_{i,j}, k=1}^{n_j} \|x_{i,k'} - x_{i,l'}\|^2 \tag{11}$$

After *kPose* computation from each action, the *kPose*s which can represent such action are selected, however there is possibility that the *kPose*s from different actions are similar which deprives of the discriminative and decreases the action recognition rate. As feature selection in [20], *kPose* is apt to discarded if there is similar one. The similarity of a *kPose* pair (y_i, y_j), where $y_i, y_j \in Y$, is defined as the the linear correlation coefficient r as below:

$$r(y_i, y_j) = \frac{\sum_k (y_{i,k} - \overline{y_i})(y_{j,k} - \overline{y_j})}{\sqrt{\sum_k (y_{i,k} - \overline{y_i})^2} \sqrt{\sum_k (y_{j,k} - \overline{y_j})^2}} \tag{12}$$

where $\overline{y_i}$ is the means of y_i, and $\overline{y_j}$ is the mean of y_j. The value of r lies between -1 and 1, inclusive. If y_i and y_j are completely correlated, r takes the value of 1 or -1; if y_i and y_j are totally independent, r is zero. It is a symmetrical measure for two *kPose*s. $\eta \in [0.8, 1]$ is a threshold, and $|r| > \eta$ means two *kPose*s are too similar to abandon one. The *kPose* computation and selection algorithm is illustrated in Table 2.

4 Action Classification

Given a new sequence to be classified and global set of *kPose*s Y, we first compute the nearest *kPose* for each frame, and construct a matrix to record the result. The index is the *kPose* and each record indicates the number of poses, which take such *kPose* as the nearest one. For *kPose* description Y_i of action X_i, we summarize the corresponding number of poses and classify the action to the one with maximal summation. The details of the proposed action classification method is given in Table 3.

The proposed classification algorithm holds high fault tolerance of pose detection rate. Assuming there are n poses taking *kPose* y_i as the nearest one, making wrong decision about a *kPose* needs $[\frac{n}{2}]+1$ error. And the probability of this situation is: $\sum_{i=[\frac{n}{2}]+1}^{n} C_n^i \alpha^{(n-i)}(1-\alpha)^i$, where α is the pose detection rate.

Table 3. The algorithm of action classification

Input: $Y = \{y_1, y_2, \cdots, y_n\}$ //global *kPose* set
$Y_i = \{y'_1, y'_2, \cdots, y'_{n_i}\}$ //*kPose*s for each action X_i
$P = \{p_1, p_2, \cdots, p_m\}$ //a novel action sequence
Output: T //the type of the novel action

```
1.  begin
2.      for j = 1 to m do begin
3.          compute the nearest kPose y_i for p_j
4.          M(y_i) ← M(y_i) + 1        // matrix M record the number
5.      end                            //of poses for each kPose
6.      for l = 1 to K do begin
7.          for k = 1 to n_l do begin
8.              sum_l ← M(y_k) + sum_l
9.          end                        //for on line 7
10.     end                            //for on line 6
11.     T = arg max_{l=1}^{K} sum_l
12. end
```

For instance, assuming $\alpha = 0.8, n = 20$, the probability of making wrong decision is 0.26%. Then, assuming there are 5 *kPose*s for an action, the probability of making wrong decision on the action class is $1 - (1 - 0.26\%)^5 = 1.29\%$.

5 Experiment

We evaluated our approach on Weizmenn data set [1] in terms of the procedure of *kPose* computation, action recognition rate, and parameter of the distribution of intra- and inter-class distance. This data set consists of ten actions performed by nine subjects. There are a total of 93 videos and 5600 frames.

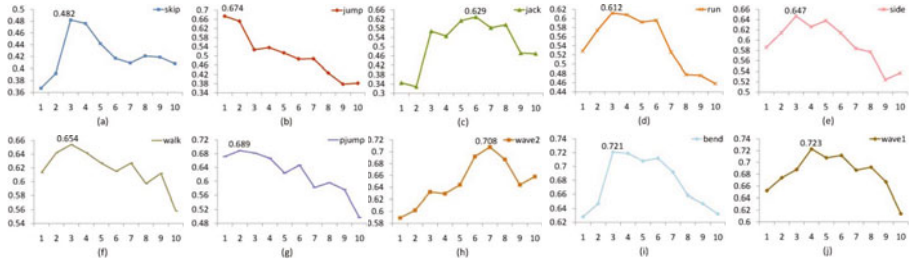


Fig. 4. The procedure of clustering an action into classes of poses: (a) skip, (b) jump, (c) jack, (d) run, (e) side, (f) walk, (g) pjump, (h) wave2, (i) bend, (j) wave1

We depict the procedures of 10 times clustering and the estimate value of different class numbers for each action in Fig. 4. The class number with maximal estimate value is selected as the *kPose* number. The number for each action are: *bend* 3 *kPose*, *jack* 6 *kPose*, *jump* 1 *kPose*, *skip* 2 *kPose*, *pjump* 2 *kPose*,

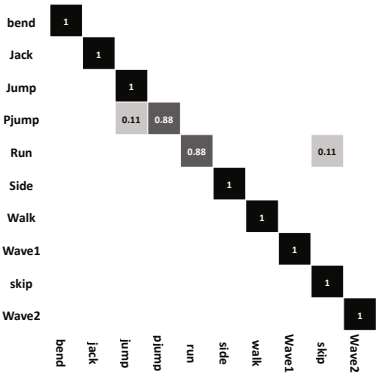
run 3 *kPose*, *side* 3 *kPose*, *walk* 3 *kPose*, *wave1* 4 *kPose* and *wave2* 7 *kPose*. The total of 34 *kPose*s are computed for different actions. Finally 28 *kPose*s constitute the global set after further selection as described in Section 3.3. There are two or three actions sharing the same *kPose*: *kPose* $\#5 \in jack, side, pjump$, *kPose* $\#25 \in run, skip, walk$, *kPose* $\#3 \in run, skip$, and *kPose* $\#19 \in run, side$.

Similar to our work, [5,6,7,8,9,10] use parts of the entire videos for action recognition on Weizmann data set. As show in Table 5a, the proposed method yields competitive results. Specially, using only 28 *kPose* for 10 actions, average 2.8 *kPose* per action, recognition result of 97.6% are obtained.

Fig. 5b shows the confusion matrix of recognition result. Our approach perfectly classifies 8 of all the actions while only mis-classifying two examples in the remaining 2 actions. The *kPose* for *jump* is $\{\#16\}$ and the *kPose* for *pjump* is $\{\#28, \#5\}$. There isn't the same *kPose* for both actions, but one *kPose* for *jump* makes the recognition less robust.

Method	Rec.rate(%)	#frames
Ours	97.6	Average 2.8
Roth [17]	94.2	2
Mauthner [14]	91.3	2
	94.3	6
Thurau[19]	70.4	1
	94.4	all
Niebles[15]	55.0	1
	72.8	all
Schindler[18]	93.5	2
	96.6	3
	99.6	10
Hatun[9]	92	half
Blank[2]	100	all
Fathi[8]	100	all

(a)Recognition rates and number of required frames for different approaches.

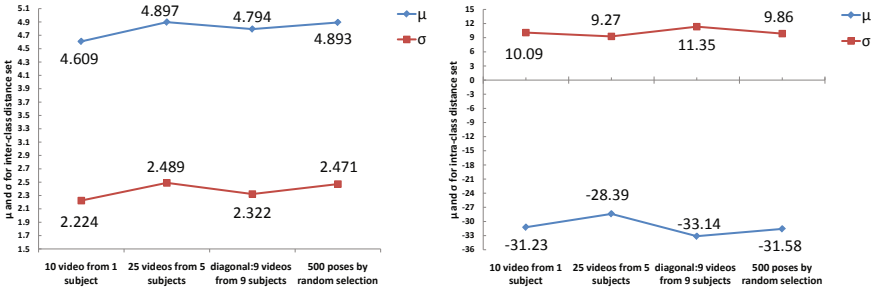


(b)The confusion matrix of proposed *kPose* method.

Fig. 5. We performed leave-one-person-out experiment. For each action, eight subjects for action training and one subject for action classification. In Table 5a, the results [1,5,6,7,8,9,10,21] are copied from the original papers.

To train the PWDM model, pose templates are selected from the data set. And the rest actions are used in action recognition. To prevent the distribution of the intra- and inter-class distance from being affected by different poses, we test four different schemes: 10 videos from one subject; 25 videos from 5 subjects, 5 for each one; 9 videos from 9 subjects like a diagonal matrix; randomly selecting 500 frames of poses (of 5600 frames at all, about 10% of the entire data set). We calculate $\mu_d, \mu_w, \sigma_d, \sigma_w$ for the four schemes, and depict in Fig. 6.

Our experiment is based on 500 random poses for distance distribution computation. Because this selection method made the least influence over the actions used in recognition part. All evaluations on this data set are performed with



(a)μ_b and σ_b (b)μ_w and σ_w

Fig. 6. Evaluation on the distribution of intra- and inter-class distance. The distribution function is trained under four different methods of pose template selection. The expected value and variance preserve slight float around the average value. The variance of these parameter: μ_b is 0.4%, σ_b is 0.7%, μ_w is 13%, σ_w is 8% corresponding to the average of each parameter.

Leave-one-person-out cross-validation, i.e., 8 subjects are used for training, the remaining one for testing; and this procedure is repeated for all 9 permutations.

6 Conclusion

We present a new action representation method for human action recognition. *KPose*s, the representative poses for each action, are extracted to describe the actions under pose-weighted distribution model which is leaned on pose templates. Corresponding classification method is carried out in a straight forward way with the *kPose* based action representation. Experimental results validated the proposed method by remarkable recognition accuracy on a benchmark data set.

The experiments presented indicate that very few poses already contains sufficient information about action. However, the HoG extracted from each frame still not enough to present the discriminative information of poses. We believe that more information obtained by additional feature could have better results. It is a beneficial attempt to extend the *kPose* concept to other fields, such as irregular action detection and complex activities recognition.

Acknowledgement. This paper is supported by the National Natural Science Foundation of China under Grant 60873124, by the Natural Science Foundation of Zhejiang Province under Grant Y1090516, and by the Fundamental Research Funds for the Central Universities under Grant 2009QNA5015.

References

1. Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. In: Tenth IEEE International Conference on Computer Vision (2005)
2. Davis, J., Bobick, A.: The representation and recognition of action using temporal templates. In: IEEE Conference on Computer Vision and Pattern Recognition (1997)

3. Efros, A., Berg, A., Mori, G., Malik, J.: Recognizing action at a distance. In: Ninth IEEE International Conference on Computer Vision (2007)
4. Yamato, J., Ohya, J., Ishii, K.: Recognizing human action in time-sequential images using hidden Markov model. In: Proc. Comp. Vis. and Pattern Rec., pp. 379–385 (1992)
5. Hatun, K., Duygulu, P.: Pose sentences: A new representation for action recognition using sequence of pose words. In: 19th International Conference on Pattern Recognition (2008)
6. Mauthner, T., Roth, P., Bischof, H.: Instant action recognition. In: Salberg, A.-B., Hardeberg, J.Y., Jenssen, R. (eds.) SCIA 2009. LNCS, vol. 5575, pp. 1–10. Springer, Heidelberg (2009)
7. Niebles, J., Fei-Fei, L.: A hierarchical model of shape and appearance for human action classification. In: IEEE Conference on Computer Vision and Pattern Recognition (2007)
8. Roth, P., Mauthner, T., Khan, I., Bischof, H.: Efficient Human Action Recognition by Cascaded Linear Classification (2010)
9. Schindler, K., Van Gool, L.: Action Snippets: How many frames does human action recognition require? In: IEEE Conference on Computer Vision and Pattern Recognition (2008)
10. Thurau, C., Hlavác, V.: Pose primitive based human action recognition in videos or still images. In: IEEE Conference on Computer Vision and Pattern Recognition (2008)
11. Dalai, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Conference on Computer Vision and Pattern Recognition (2005)
12. 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)
13. Laptev, I., Lindeberg, T.: SpaceCtime interest points. In: Tenth IEEE International Conference on Computer Vision (2003)
14. Harris, C., Stephens, M.: A combined corner and edge detector. In: Alvey Vision Conference, p. 50 (1988)
15. Dollar, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (2005)
16. Danafar, S., Gheissari, N.: Action recognition for surveillance applications using optic flow and SVM. In: Yagi, Y., Kang, S.B., Kweon, I.S., Zha, H. (eds.) ACCV 2007, Part II. LNCS, vol. 4844, pp. 457–466. Springer, Heidelberg (2007)
17. Lu, W., Little, J.: Simultaneous tracking and action recognition using the pca-hog descriptor. In: Canadian Conference on Computer and Robot Vision (2006)
18. Patron-Perez, A., Reid, I.: A probabilistic framework for recognizing similar actions using spatio-temporal features. In: British Machine Vision Conference
19. Likas, A., Vlassis, N., et al.: The global k-means clustering algorithm. Pattern Recognition, 451–461 (2003)
20. Yu, L., Liu, H.: Feature selection for high-dimensional data: A fast correlation-based filter solution. Machine Learning-International, 856 (2003)
21. Fathi, A., Mori, G.: Action recognition by learning mid-level motion features. In: IEEE Conference on Computer Vision and Pattern Recognition (2008)

Identifying Surprising Events in Videos Using Bayesian Topic Models

Avishai Hendel, Daphna Weinshall, and Shmuel Peleg

School of Computer Science, The Hebrew University, Jerusalem, Israel

Abstract. Automatic processing of video data is essential in order to allow efficient access to large amounts of video content, a crucial point in such applications as video mining and surveillance. In this paper we focus on the problem of identifying interesting parts of the video. Specifically, we seek to identify atypical video events, which are the events a human user is usually looking for. To this end we employ the notion of Bayesian surprise, as defined in [1,2], in which an event is considered surprising if its occurrence leads to a large change in the probability of the world model. We propose to compute this abstract measure of surprise by first modeling a corpus of video events using the Latent Dirichlet Allocation model. Subsequently, we measure the change in the Dirichlet prior of the LDA model as a result of each video event's occurrence. This change of the Dirichlet prior leads to a closed form expression for an event's level of surprise, which can then be inferred directly from the observed data. We tested our algorithm on a real dataset of video data, taken by a camera observing an urban street intersection. The results demonstrate our ability to detect atypical events, such as a car making a U-turn or a person crossing an intersection diagonally.

1 Introduction

1.1 Motivation

The availability and ubiquity of video from security and monitoring cameras has increased the need for automatic analysis and classification. One urging problem is that the sheer volume of data renders it impossible for human viewers, the ultimate classifiers, to watch and understand all of the displayed content. Consider for example a security officer who may need to browse through the hundreds of cameras positioned in an airport, looking for possible suspicious activities - a laborious task that is error prone, yet may be life critical. In this paper we address the problem of unsupervised video analysis, having applications in various domains, such as the inspection of surveillance videos, examination of 3D medical images, or cataloging and indexing of video libraries.

A common approach to video analysis serves to assist human viewers by making video more accessible to sensible inspection. In this approach the human judgment is maintained, and video analysis is used only to assist viewing. Algorithms have been devised to create a compact version of the video, where only

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 448–459, 2011.

certain activities are displayed [3], or where all activities are displayed using video summarization [4].

We would like to go beyond summarization; starting from raw video input, we seek an automated process that will identify the unusual events in the video, and reduce the load on the human viewer. This process must first extract and analyze activities in the video, followed by establishing a model that characterizes these activities in a manner that permits meaningful inference. A measure to quantify the significance of each activity is needed as a last step.

1.2 Related Work

Boiman and Irani [3] propose to recognize irregular activities in video by considering the complexity required to represent the activity as a composition of codebook video patches. This entails dense sampling of the video and is therefore very time consuming, making it cumbersome to apply this algorithm to real world data. Itti and Baldi [1] present a method for surprise detection that operates in low-level vision, simulating early vision receptors. Their work is directed at the modeling and prediction of human visual attention, and does not address the understanding of high level events.

Other researchers use Bayesian topic models as a basis for the representation of the environment and for the application of inference algorithms. To detect landmark locations Ranganathan and Dellaert [5] employ the surprise measure over an appearance place representation. Their use of only local shape features makes their approach applicable in the field of topological mappings, but not in object and behavior based video analysis. Two closely related models are that of Hospedales et. al. [6] and Wang et. al. [7]. Both models use topic models over low level features to represent the environment. [6] uses Bayesian saliency to recognize irregular patterns in video scenes, while [7] defines abnormal events as events with low likelihood. Both approaches may be prone to the 'white snow paradox' [1], where data that is more informative in the classic Shannon interpretation does not necessarily match human semantic interests.

1.3 Our Approach

We present a generative probabilistic model that accomplishes the tasks outlined above in an unsupervised manner, and test it in a real world setting of a webcam viewing an intersection of city streets.

The preprocessing stage consists of the extraction of video activities of high level objects (such as vehicles and pedestrians) from the long video streams given as input. Specifically, we identify a set of video events (video tubes) in each video sequence, and represent each event with a 'bag of words' model. In previous work words were usually chosen to be local appearance features, such as SIFT [8,9] or spatio-temporal words [10]. We introduce the concept of 'transition words', which allows for a compact, discrete representation of the dynamics of an object in a video sequence. Despite its simplicity, this representation is successful in capturing the essence of the input paths. The detected activities

are then represented using a latent topic model, a paradigm that has already shown promising results [6,9,11,12].

Next, we examine the video events in a rigorous Bayesian framework, to identify the most interesting events present in the input video. Thus, in order to differentiate intriguing events from the typical commonplace events, we measure the effect of each event on the observer's beliefs about the world, following the approach put forth in [1,2]. We propose to measure this effect by comparing the prior and posterior parameters of the latent topic model, which is used to represent the overall data. We then show that in the street camera scenario, our model is able to pick out atypical activities, such as vehicle U-turns or people walking in prohibited areas.

The rest of the paper is organized as follows: in Section 2 we describe the basic extraction and representation of activities in input videos. In Section 3 the 'bag of words' model is used to represent the input in a statistical generative manner as explained above. Section 4 and Section 5 introduce the Bayesian framework for identifying atypical events, and in Section 6 the application of this framework to real world data is presented.

2 Activity Representation

2.1 Objects as Space Time Tubes

To recognize unusual activities in input videos, we first need to isolate and localize objects out of the image sequence. The fundamental representation of objects in our model is that of 'video tubes' [13]. A tube is defined by a sequence of object masks carved through the space time volume, assumed to contain a single object of interest (e.g., in the context of street cameras, it may be a vehicle or a pedestrian). This localizes events in both space and time, and enables the association of local visual features with a specific object, rather than an entire video.

Tubes are extracted by first segmenting each video frame into background and foreground regions, using a modification of the 'Background Cut' method, described in [14]. Foreground blobs from consecutive frames are then matched by spatial proximity to create video tubes that extend through time. A collection of tubes extracted from an input video sequence is the corpus used as the basis for later learning stages.

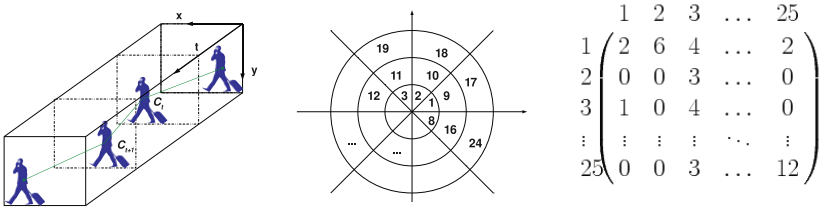
2.2 Trajectories

An obvious and important characteristic of a video tube is its trajectory, as defined by the sequence of its spatial centroids. Encoding the dynamics of an object is a crucial step for successful subsequent processing. A preferable encoding in our setting should capture the characteristic of the tube's path in a compact and effective way, while considering location, speed and form.

Of the numerous existing approaches, we use a modification of the method suggested in [15]. Denote the displacement vector between two consecutive spatial centroids C_t and C_{t+1} as $D = \overrightarrow{C_t C_{t+1}}$ (Fig. 1a). Since the temporal difference

is constant (a single frame interval between centroids) we may ignore it, and assume D has only spatial components $(\Delta x, \Delta y)$. Quantization of possible values of D is obtained through the following procedure: First, the magnitude of all displacement vectors is normalized by the largest displacement found in the trajectory - $\|D\|_{max}$. Then the normalized magnitude is assigned to one of three uniform quantization levels. The orientation component of each displacement vector is binned into one of eight sectors of the unit circle, each sector covering $\pi/4$ radians. The combination of three magnitude scales and eight orientation sectors gives 24 quantization bins (Fig. 1b). Adding another bin to indicate zero displacement, we have a total of 25 displacement bins. After quantizing all of the displacement vectors of a trajectory, we create a transition occurrence matrix (Fig. 1c), indicating the frequency of bin transitions in the tube.

This matrix can be viewed as a histogram of 'transition words', where each word describes the transition between two consecutive quantized displacement vectors. The final representation of a trajectory is this histogram, indicating the relative frequency of the 625 possible transitions.



(a) (b) (c)

Fig. 1. Trajectory representation: the three stages of our trajectory representation: (a) compute the displacement of the centroids of the tracked object between frames, (b) quantize each displacement vector into one of 25 quantization bins, and (c) count the number of different quantization bin transitions in the trajectory into a histogram of bin transitions

3 Modeling of Typical Activities Using LDA

The *Latent Dirichlet Allocation* (LDA) model is a generative probabilistic model, first introduced in the domain of text analysis and classification [16]. As other topic models, it aims to discover latent topics whose mixture is assumed to be the underlying cause of the observed data. Its merits lie in that it is a truly generative model that can be learned in a completely unsupervised manner, it allows the use of priors in a rigorous Bayesian manner, and it does not suffer from over-fitting issues like its closely related pLSA model [17]. It has been successfully applied recently to computer vision tasks, where the text topics have been substituted with scenery topics [9] or human action topics [12].

As is common with models from the 'bag of words' paradigm, the entities in question (video tubes, in our case) are represented as a collection of local, discrete features. The specific mixture of topics of a single video tube determines the observed distribution of these features.

More formally, assume we have gathered a set of video tubes and their trajectories in the corpus $T = \{T_1, T_2, ..., T_m\}$. Each tube is represented as a histogram of transition words taken from the trajectory vocabulary $V = \{w_{1-1}, w_{1-2}, ..., w_{25-24}, w_{25-25}\}, |V| = 625$. Thus the process that generates each trajectory T_j in the corpus is:

1. Choose $N \sim Poisson(\xi)$, the number of feature words (or, in effect, the length of the trajectory).
2. Choose $\theta \sim Dirichlet(\alpha)$, the mixture of latent topics in this tube.
3. For each of the N words w_n, where $1 \leq n \leq N$:
   - Choose a topic $z_n \sim Multinomial(\theta)$
   - Choose a codebook word w_n from the multinomial distribution $p(w_n \mid z_n, \beta)$

In this model, α is a k-dimensional vector that is the parameter for the Dirichlet distribution, k is the predetermined number of hidden topics, and β is a $k \times V$ matrix that characterizes the word distributions conditioned on the selected latent topic. The entry $\beta_{i,j}$ corresponds to the measure $p(w^j = 1 \mid z^i = 1)$. A plate notation representation of the model is shown in Fig. 2a. The joint distribution of the trajectory topic mixture θ, the set of transition words $\mathbf{w}$ and their corresponding topics $\mathbf{z}$ can be summarized as:

$$p(\theta, \mathbf{w}, \mathbf{z} \mid \alpha, \beta) = p(\theta \mid \alpha) \prod_{n=1}^{N} p(z_n \mid \theta) p(w_n \mid z_n, \beta) \tag{1}$$

Once the model has been learned and the values of the vector α and the matrix β are known, we can compute the posterior distribution of the hidden variables of a new unseen tube:

$$p(\theta, \mathbf{z} \mid \mathbf{w}, \alpha, \beta) = \frac{p(\theta, \mathbf{w}, \mathbf{z} \mid \alpha, \beta)}{p(\mathbf{w} \mid \alpha, \beta)} \tag{2}$$

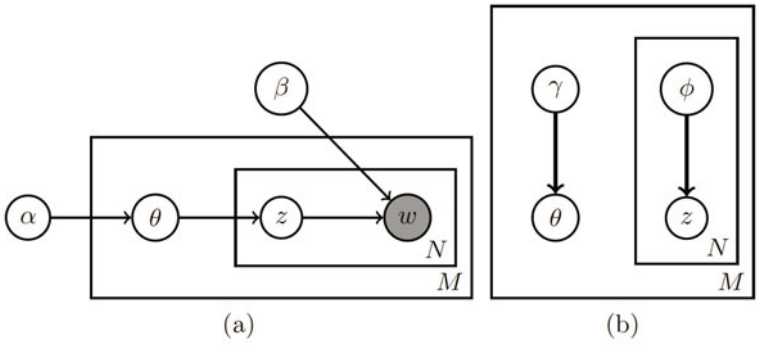


Fig. 2. (a) Graphical model represntation of LDA using plate notation. (b) Simplified model used to approximate the posterior distribution.

Although this distribution is computationally intractable, approximate inference algorithms such as Gibbs sampling or variational methods can be used. The basic principle behind the variational approach [18] is to consider a simplified graphical model, where problematic ties between variables are removed. The edges between θ, $\mathbf{z}$, and $\mathbf{w}$ cause the coupling between θ and β, which is the reason for the intractibility of Eq. (2). Dropping these edges and incorporating the free variational parameters γ and ϕ into the simplified model (Fig. 2b), we acquire a family of distributions on the latent variables that is tractable:

$$q(\theta, \mathbf{z} \mid \gamma, \phi) = q(\theta \mid \gamma) \prod_{n=1}^{N} q(z_n \mid \phi_n) \tag{3}$$

where γ approximates the Dirichlet parameter α and ϕ mirrors the multinomial parameters β.

Now an optimization problem can be set up to minimize the difference between the resulting variational distribution and the true (intractable) posterior, yielding the optimizing parameters (γ^*, ϕ^*), which are a function of $\mathbf{w}$. The Dirichlet parameter $\gamma^*(\mathbf{w})$ is the representation of the new trajectory in the simplex spanned by the latent topics. Thus it characterizes the composition of the actual path out of the k basic trajectory topics.

Based on this inference method, Blei [16] suggests an alternating variational EM procedure to estimate the parameters of the LDA model:

1. E-Step: For each tube, find the optimizing values of the variational parameters $\{\gamma_t^*, \phi_t^* : t \in T.\}$
2. M-Step: Maximize the resulting lower bound on the log likelihood of the entire corpus with respect to the model parameters α and β.

The estimation of the model's parameters α and β completes our observer's model of its world. The Dirichlet prior α describes the common topic mixtures that are to be expected in video sequences taken from the same source as the training corpus. A specific mixture θ_t determines the existence of transitions found in the trajectory using the per-topic word distribution matrix β. Crude classification of tubes into one of the learned latent topics can be done simply by choosing the topic that corresponds to the maximal element in the posterior Dirichlet parameter γ_t^*.

4 Surprise Detection

The notion of surprise is, of course, human-centric and not well defined. Surprising events are recognized as such with regard to the domain in question, and background assumptions that can not always be made explicit. Thus, rule based methods that require manual tuning may succeed in a specific setting, but are doomed to failure in less restricted settings. Statistical methods, on the other hand, require no supervision. Instead, they attempt to identify the expected events from the data itself, and use this automatically learned notion of typicality to recognize the extraordinary events.

Such framework is proposed in the work by Itti [1] and Schmidhuber [2]. Dubbed 'Bayesian Surprise', the main conjecture is that a surprising event from the viewpoint of an observer is an event that modifies its current set of beliefs about the environment in a significant manner. Formally, assume an observer has a model M to represent its world. The observer's belief in this model is described by the prior probability of the model $p(M)$ with regard to the entire model space $\mathcal{M}$. Upon observing a new measurement t, the observer's model changes according to Bayes' Law:

$$p(M \mid t) = \frac{p(M)p(t \mid M)}{p(t)} \tag{4}$$

This change in the observer's belief in its current model of the world is defined as the surprise experienced by the observer. Measurements that induce no or minute changes are not surprising, and may be regarded as 'boring' or 'obvious' from the observer's point of view. To quantify this change, we may use the KL divergence between the prior and posterior distributions over the set $\mathcal{M}$ of all models:

$$S(t, M) = KL(p(M), p(M \mid t)) = \int_{\mathcal{M}} p(M) log \frac{p(M)}{p(M \mid t)} dM \tag{5}$$

This definition is intuitive in that surprising events that occur repeatedly will cease to be surprising, as the model is evolving. The average taken over the model space also ensures that events with very low probability will be regarded as surprising only if they induce a meaningful change in the observer's beliefs, thus ignoring noisy incoherent data that may be introduced.

Although the integral in Eq. (5) is over the entire model space, turning this space to a parameter space by assuming a specific family of distributions may allow us to compute the surprise measure analytically. Such is the case with the Dirichlet family of distributions, which has several well known computational advantages: it is in the exponential family, has finite dimensional sufficient statistics, and is conjugate to the multinomial distribution.

5 Bayesian Surprise and the LDA Model

As noted above, the LDA model is ultimately represented by its Dirichlet prior α over topic mixtures. It is a natural extension now to apply the Bayesian surprise framework to domains that are captured by LDA models.

Recall that video tubes in our 'bag of words' model are represented by the posterior optimizing parameter γ^*. Furthermore, new evidence also elicits a new Dirichlet parameter for the world model of the observer, $\widehat{\alpha}$. To obtain $\widehat{\alpha}$, we can simulate one iteration of the variational EM procedure used above in the model's parameters estimation stage, where the word distribution matrix β is kept fixed. This is the Dirichlet prior that would have been calculated had the

new tube been appended to the training corpus. The Bayesian Surprise formula when applied to the LDA model can be now written as:

$$S(\alpha, \widehat{\alpha}) = KL_{DIR}(\alpha, \widehat{\alpha}) \tag{6}$$

The Kullback - Leibler divergence of two Dirichlet distributions can be computed as [19]:

$$KL_{DIR}(\alpha, \widehat{\alpha}) = log\frac{\Gamma(\boldsymbol{\alpha})}{\Gamma(\widehat{\boldsymbol{\alpha}})} + \sum_{i=1}^{k} log\frac{\Gamma(\widehat{\alpha}_i)}{\Gamma(\alpha_i)} + \sum_{i=1}^{k} [\alpha_i - \widehat{\alpha}_i][\psi(\alpha_i) - \psi(\boldsymbol{\alpha})] \tag{7}$$

where

$$\boldsymbol{\alpha} = \sum_{i=1}^{k} \alpha_i \quad \text{and} \quad \widehat{\boldsymbol{\alpha}} = \sum_{i=1}^{k} \widehat{\alpha}_i$$

and Γ and ψ are the gamma and digamma functions, respectively.

Thus each video event is assigned a surprise score, which reflects the tube's deviation from the expected topic mixture. In our setting, this deviation may correspond to an unusual trajectory taken by an object, such as 'car doing a U-turn', or 'person running across the road'. To obtain the most surprising events out of a corpus, we can select those tubes that receive a surprise score that is higher than some threshold.

6 Experimental Results

6.1 Dataset

Surveillance videos are a natural choice to test and apply surprise detection algorithms. Millions of cameras stream endless videos that are notoriously hard to monitor, where significant events can be easily overlooked by an overwhelmed human observer. We test our model on data obtained from a real world street camera, overlooking an urban road intersection. This scenario usually exhibits structured events, where pedestrians and vehicles travel and interact in mostly predefined ways, constrained by the road and sidewalk layout. Aside from security measures, intersection monitoring has been investigated and shown to help in reducing pedestrian and vehicle conflicts, which may result in injuries and crashes [20].

The training input sequence consists of an hour of video footage, where frame resolution is 320x240 and the frame rate is 10fps. The test video was taken in the subsequent hour. The video was taken during the morning, when the number of events is relatively small. Still, each hour contributed about 1000 video tubes. The same intersection at rush hours poses a significant challenge to the tracking algorithm due to multiple simultaneous activities and occlusions, but this tracking is not the focus of this work. Subsequent analysis is agnostic to the mode of tube extraction, and the method we used can be easily replaced by any other method.

6.2 Trajectory Classification

The first step in our algorithm is the construction of a model that recognizes typical trajectories in the input video. We fix k, the number of latent topics to be 8. Fig. 3 shows several examples of classified objects from four of the eight model topics, including examples from both the training and test corpora. Fig. 4 shows the distribution of trajectories into topics, in the train and test corpora.

Note that some of the topics seem to have a semantic meaning. Thus, on the basis of trajectory description alone, our model was able to automatically catalog the video tubes into semantic movement categories such as 'left to right', or 'top to bottom', with further distinction between smooth constant motion (normally cars) and the more erratic path typically exhibited by people. It should be noted, however, that not all latent topics correspond with easily interpretable patterns of motion as depicted in Fig. 3. Other topics seem to capture complicated path forms, where pauses and direction changes occur, with one topic representing 'standing in place' trajectories.

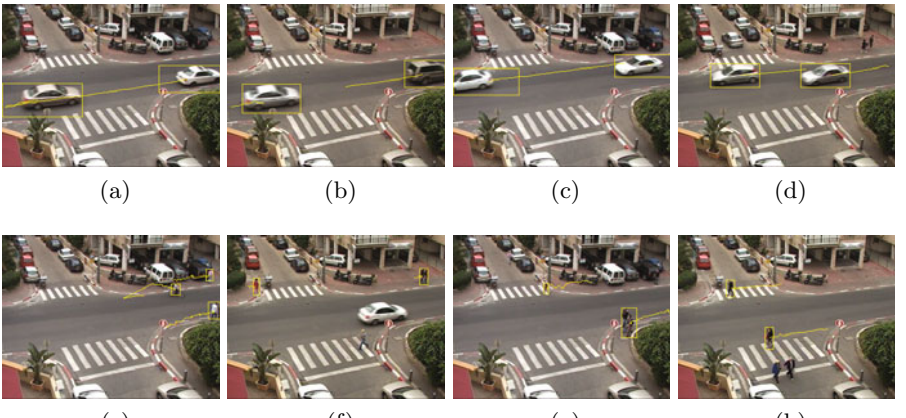


Fig. 3. Trajectory classifications: (a,b) cars going left to right, (c,d) cars going right to left, (e,f) people walking left to right, and (g,h) people walking right to left

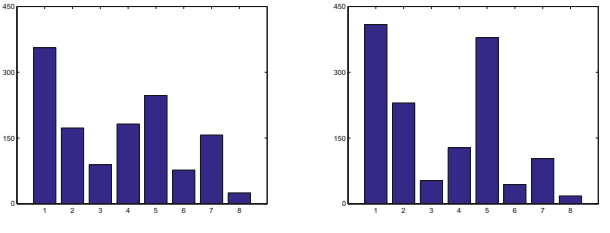
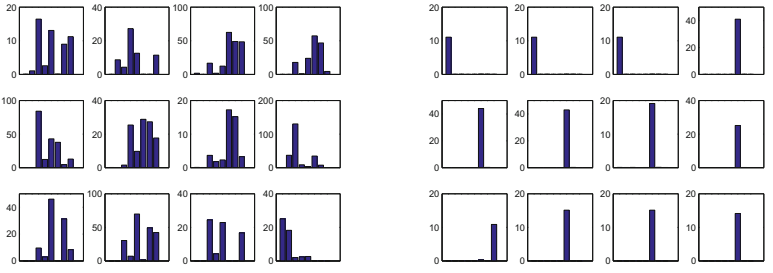


Fig. 4. Number of trajectories assigned to each topic in the train (left) and test (right) corpora. 1306 tubes were extracted from the training sequence, and 1364 from the test sequence.

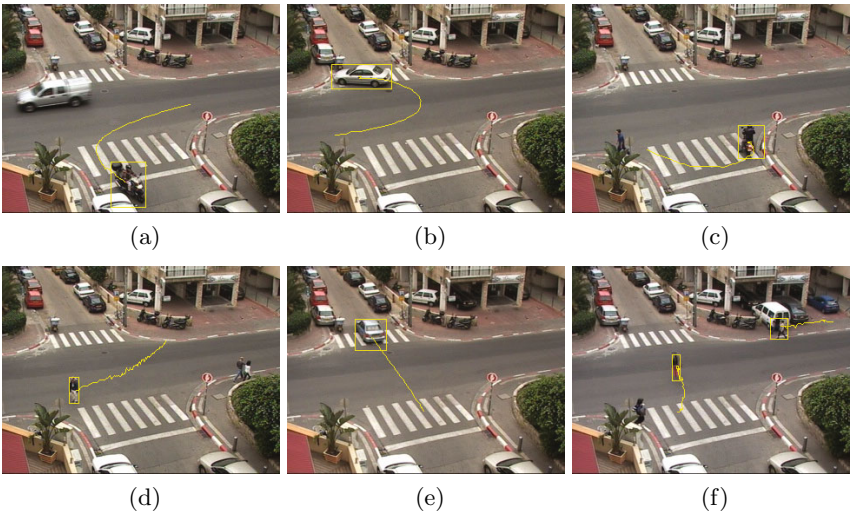
6.3 Surprising Events

To identify the atypical events in the corpus, we look at those tubes which have the highest surprise score. Several example tubes which fall above the 95th percentile are shown in Fig. 6. They include such activities as a vehicle performing a U-turn, or a person walking in a path that is rare in the training corpus, like crossing the intersection diagonally.



(a) (b)

Fig. 5. Posterior Dirichlet parameters γ^* values for the most surprising (a) and typical (b) events. Each plot shows the values of each of the $k = 8$ latent topics. Note that the different y scales correspond to different trajectory lengths (measured in frames).



(a) (b) (c)

(d) (e) (f)

Fig. 6. Surprising events: (a) a bike turning into a one-way street from the wrong way, (b) a car performing a U-turn, (c) a bike turning and stalling over pedestrian crossing, (d) a man walking across the road, (e) a car crossing the road from bottom to top, (f) a woman moving from the sidewalk to the middle of the intersection.

In Fig. 5 the γ^* values of the most surprising and typical trajectories are shown. It may be noted that while 'boring' events generally fall into one of the learned latent topics exclusively (Fig. 5b), the topic mixture of surprising events has massive counts in several topics at once (Fig. 5a). This observation is verified by computing the mean entropy measure of the γ^* parameters, after being normalized to a valid probability distribution:

$$\overline{H}(\gamma_{surprising}) = 1.2334, \quad \overline{H}(\gamma_{typical}) = 0.5630$$

7 Conclusions

In this work we presented a novel integration between the generative probabilistic model LDA and the Bayesian surprise framework. We applied this model to real world data of urban scenery, where vehicles and people interact in natural ways. Our model succeeded in automatically obtaining a concept of the normal behaviors expected in the tested environment, and in applying these concepts in a Bayesian manner to recognize those events that are out of the ordinary. Although the features used are fairly simple (the trajectory taken by the object), complex surprising events such as a car stalling in its lane, or backing out of its parking space were correctly identified, judged against the normal paths present in the input.

References

1. Itti, L., Baldi, P.: A principled approach to detecting surprising events in video. In: CVPR, vol. (1), pp. 631–637 (2005)
2. Schmidhuber, J.: Driven by compression progress: A simple principle explains essential aspects of subjective beauty, novelty, surprise, interestingness, attention, curiosity, creativity, art, science, music, jokes. In: Pezzulo, G., Butz, M.V., Sigaud, O., Baldassarre, G. (eds.) Anticipatory Behavior in Adaptive Learning Systems. LNCS, vol. 5499, pp. 48–76. Springer, Heidelberg (2009)
3. Boiman, O., Irani, M.: Detecting irregularities in images and in video. International Journal of Computer Vision 74, 17–31 (2007)
4. Pritch, Y., Rav-Acha, A., Peleg, S.: Nonchronological video synopsis and indexing. IEEE Trans. Pattern Anal. Mach. Intell. 30, 1971–1984 (2008)
5. Ranganathan, A., Dellaert, F.: Bayesian surprise and landmark detection. In: ICRA, pp. 2017–2023 (2009)
6. Hospedales, T., Gong, S., Xiang, T.: A markov clustering topic model for mining behaviour in video. In: ICCV (2009)
7. Wang, X., Ma, X., Grimson, E.: Unsupervised activity perception by hierarchical bayesian models. In: CVPR (2007)
8. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60, 91–110 (2004)
9. Fei-Fei, L., Perona, P.: A bayesian hierarchical model for learning natural scene categories. In: CVPR, vol. (2), pp. 524–531 (2005)

10. Laptev, I., Lindeberg, T.: Local descriptors for spatio-temporal recognition. In: MacLean, W.J. (ed.) SCVMA 2004. LNCS, vol. 3667, pp. 91–103. Springer, Heidelberg (2006)
11. Sivic, J., Russell, B.C., Efros, A.A., Zisserman, A., Freeman, W.T.: Discovering objects and their localization in images. In: ICCV, pp. 370–377 (2005)
12. Niebles, J.C., Wang, H., Fei-Fei, L.: Unsupervised learning of human action categories using spatial-temporal words. International Journal of Computer Vision 79, 299–318 (2008)
13. Pritch, Y., Ratovitch, S., Hendel, A., Peleg, S.: Clustered synopsis of surveillance video. In: AVSS, pp. 195–200 (2009)
14. Sun, J., Zhang, W., Tang, X., Shum, H.-Y.: Background cut. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 628–641. Springer, Heidelberg (2006)
15. Sun, J., Wu, X., Yan, S., Cheong, L.F., Chua, T.S., Li, J.: Hierarchical spatio-temporal context modeling for action recognition. In: CVPR, pp. 2004–2011 (2009)
16. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. In: NIPS, pp. 601–608 (2001)
17. Hofmann, T.: Probabilistic latent semantic analysis. In: UAI, pp. 289–296 (1999)
18. Jordan, M.I., Ghahramani, Z., Jaakkola, T., Saul, L.K.: An introduction to variational methods for graphical models. Machine Learning 37, 183–233 (1999)
19. Penny, W.D.: Kullback-liebler divergences of normal, gamma, dirichlet and wishart densities. Technical report, Wellcome Department of Cognitive Neurology (2001)
20. Hughes, R., Huang, H., Zegeer, C., Cynecki, M.: Evaluation of automated pedestrian detection at signalized intersections (2001)

Face Detection with Effective Feature Extraction

Sakrapee Paisitkriangkrai[1,2,*], Chunhua Shen[3], and Jian Zhang[1,3]

[1] The University of New South Wales
[2] The University of Adelaide
paul.paisitkriangkrai@adelaide.edu.au
[3] National ICT Australia
{chunhua.shen,jian.zhang}@nicta.com.au

Abstract. There is an abundant literature on face detection due to its important role in many vision applications. Since Viola and Jones proposed the first real-time AdaBoost based face detector, Haar-like features have been adopted as the method of choice for frontal face detection. In this work, we show that simple features other than Haar-like features can also be applied for training an effective face detector. Since, single feature is not discriminative enough to separate faces from difficult non-faces, we further improve the generalization performance of our simple features by introducing feature co-occurrences. We demonstrate that our proposed features yield a performance improvement compared to Haar-like features. In addition, our findings indicate that features play a crucial role in the ability of the system to generalize.

1 Introduction

Face detection is an important first step for several computer vision applications. It was not until recently that face detection problem received considerable attention among researchers owing to the impressive performance of Viola and Jones' face detector [1]. Their detector was the first algorithm that achieved real-time detection speed and high accuracy comparable to previous state of the art methods. Their work consists of three contributions. The first contribution is a cascade of classifiers. The second contribution is the boosted classifier where a combination of linear classifiers is formed to achieve fast calculation time with high accuracy. The last contribution is a simple rectangular Haar-like feature which can be extracted and computed in fewer than ten Central Processing Unit (CPU) operations using integral image.

Haar-like wavelet features are defined as a difference between the accumulated intensities of filled rectangles and unfilled rectangles. Several researchers have proposed various approaches to extend the robustness and discriminative power of Haar-like features [2,3]. Lienhart *et al.* proposed a novel set of rotated Haar-like features which can also be calculated efficiently [2]. Li and Zhang later

[*] This work was performed at The University of New South Wales.

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 460–470, 2011.

proposed a simple Haar wavelet, which separates Haar-like rectangles at some distance apart [3]. The authors tested their proposed features on multi-view faces and demonstrated excellent performance. Huang *et al.* [4] further extended Haar-like features in a slightly different way. Instead of using rectangles, they proposed sparse granular features, which represent a sum of pixel intensities in a square. An efficient weak learning algorithm is introduced which adopts heuristic search method in pursuit of discriminative sparse granular features. Since, sparse granular features have a smaller rectangular region than Haar-like features; it has a better discriminative power for multi-view faces due to their less within-class variance.

Nonetheless, Haar-like wavelet and its variants are not the only visual descriptor that has gained tremendous success, other locally extracted features, *e.g.*, edge orientation histograms (EOH) [5], Histogram of Oriented Gradients (HoG) [6], Local Binary Pattern (LBP) operator [7], have also performed remarkably well in vision applications. Levi and Weiss [5] proposed EOH which divides edges into a number of bins. Three set of features are then used to describe an image region:- a ratio between each orientation, a ratio between a single orientation and the difference between two symmetric orientations. For frontal face detection, EOH achieves state of the art performance using only a few hundred training images. Dalal and Triggs proposed histogram of oriented gradients in the context of human detection [6]. Their method uses a dense grid of histogram of oriented gradients, computed over blocks of various sizes. Ojala *et al.* proposed LBP feature, which is derived from a general definition of texture in local neighborhood [7]. Two most important properties of LBP operators are its invariance against illumination changes and its computational simplicity. Recently, Wang *et al.* [8] combined HoG and LBP descriptors as the feature set for human detection. The authors reported that the combined classifier yields the best descriptor for classifying pedestrians.

Although these recently proposed descriptors have shown excellent results in many empirical studies, when compared to simple Haar-like features, they have a *much higher complexity* and *computation time*. Since, face data sets are less complex than human data sets, *i.e.*, faces have less variation than human and partial occlusions happen less in faces, we simplify the best descriptor reported in Wang *et al.* [8], namely a combination of HoG and LBP, for the task of face detection. Our aim is to lower the time it takes to extract features while maintaining their high discriminative power. In order to further improve the generalization performance of our simple features, we create a more distinctive features combination using sparse least square regression.

The rest of the paper is organized as follows. Section 2 begins by describing the concept of HoG and LBP descriptors. We then provide details of our simple edge descriptors, which combine the strength of both HOG and LBP descriptors, and propose our joint features. Numerous experimental results are presented in Section 3. Section 4 provides a brief discussion of our proposed features. We conclude the paper in Section 5.

2 Simple Edge Descriptors

Our descriptors are based on HOG and LBP features, which have shown to give excellent results in many vision applications. The intuition of our descriptors is that the appearance of faces can be well characterized by horizontal, vertical and diagonal edges, as shown in [1]. Hence, we modify parameters' value used in HoG and LBP to reduce the feature extraction time. In this section, we first give a brief review of HoG and LBP descriptors. We then mention how we adopt HoG and LBP to face detection problem.

2.1 HoG and LBP

After Lowe proposed Scale Invariant Feature Transformation (SIFT) [9], many researchers have studied the use of orientation histograms in other areas. Dalal and Triggs [6] proposed histogram of oriented gradients in the context of human detection. Their method uses a dense grid of histogram of oriented gradients, computed over blocks of various sizes. Each block consists of a number of cells. A local 1D orientation histogram of gradients is formed from the gradient orientations of sample points within a region. Each histogram divides the gradient angle range into a predefined number of bins. The gradient magnitudes vote into the orientation histogram. In [6], each block is quantized into 2×2 cells and the gradient angle in each cell is quantized into 9 orientations (unsigned gradients, *i.e.*, $[0, 180]$ degrees), resulting in a 36-dimensional descriptor (9 bins/cell $\times$ 4 cells/block). In their approach, the final object descriptor is obtained by concatenating the orientation histograms over all blocks.

LBP was first proposed as a gray level invariant texture primitive. LBP operator describes each pixel by its relative gray level to its neighbouring pixels, *e.g.*, if the gray level of the neighbouring pixel is higher or equal, the value is set to one, otherwise to zero. Hence, each center pixel can be represented by a binary string. The histogram of binary patterns computed over a region is then used to describe image texture. Fig. 1 illustrates LBP of radius 1 pixel with 8 neighbours. For LBP of radius 1 pixel, an 8-bit binary number is generated, resulting in 2^8 distinct values for the binary pattern. LBP has several properties that favour its usage, *e.g.*, it is robust against illumination changes, has high discriminative power and also fast to compute.

2.2 Rectangular Features Based on HoG and LBP

Similar to HoG and LBP, we consider the change of pixel intensities in a small image neighbourhood to provide a measurement of local gradients inside each rectangular region. For HoG, we set the number of cells in each block to be one. Each block can have various rectangular sizes. For LBP, a binary pattern is extracted inside a given rectangular region. In this paper, we simplify the computational complexity of both HoG and LBP features for fast feature extraction time. To achieve this, we quantize the gradient angle into 2 orientations (horizontal and vertical axes). We build histogram for both signed and unsigned

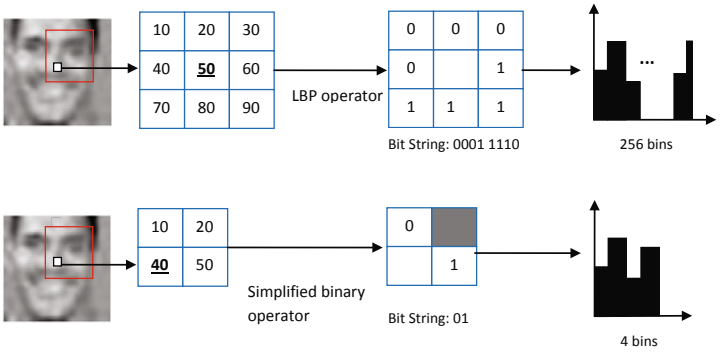


Fig. 1. *Top:* An illustration of LBP. *Bottom:* An illustration of our edge binary pattern. Both descriptors have a radius of 1 pixel.

gradients. Hence, each block can be represented by a 4-D feature vector. A vector is normalized to an ℓ_2 unit length. We also represent our features similar to LBP by making use of binary pattern on a smaller neighbourhood. Fig. 1 illustrates our simplified edge binary pattern of radius 1 pixel. For each rectangular block, we normalize HoG and LBP separately and concatenate them to get the final block descriptor. Building on the fundamental concept of HoG and LBP descriptors, the new descriptor has many invariance properties such as being tolerance to illumination changes, robustness to image noise, and computational simplicity. In our paper, multidimensional decision stumps are used as AdaBoost weak learner to train our features.

2.3 Joint Rectangular Features

In our work, we use the AdaBoost classifier with multidimensional decision stumps as weak learner. One disadvantage of training a weak learner with a single feature is that the generalization performance hardly improves in later rounds of boosting. Many researchers have observed that adding more weak learners can reduce the training error but not the generalization error [3,10,11]. More importantly, the detection performance of single features drops drastically in later stages of the cascade. We believe that single features are not discriminative enough to separate faces from difficult non-faces.

The use of feature co-occurrences in each weak classifier has been shown to yield higher classification performance compared to the use of a single feature [11]. Similar to Mita *et al.* [11], we solve this problem by applying the concept of joint features to create a more distinctive co-occurrence of features. Instead of using class-conditional joint probabilities, we approach this problem by using sparse least square regression to train our weak classifiers. Least square regression is proved to be one of the most effective weak classifiers in various literatures [12,13]. Joint features using sparse least square regression make it possible to classify difficult samples that are misclassified by weak classifiers using a single feature.

Let training data sets consist of n samples (X_i, y_i), $i = 1..n$, where $X_i = [R_1, R_2, .., R_j]$ is the vector of 1D rectangular features and y_i is an object class $\in \{-1, 1\}$. The least square model has the form $f(X, \boldsymbol{\beta}) = \beta_1 R_1 + \beta_2 R_2 + \cdots + \beta_j R_j + \beta_0$. The least square method finds optimum parameters, $\hat{\boldsymbol{\beta}}$, where the weighted sum of squared residuals, $\sum_{i=1}^{n} w_i [y_i - f(X_i, \hat{\boldsymbol{\beta}})]^2$, is minimized. Here, w_i is the sample weights. In order to construct a set of distinctive feature co-occurrences, we focus on a subset of rectangular features. In other words, we add a sparsity constraint into our least square problem. The optimization problem can now be defined as

$$\begin{aligned} \min_{i} \quad & \sum_{i} w_i [y_i - f(X_i, \hat{\boldsymbol{\beta}})]^2, \\ \text{subject to} \quad & \mathbf{Card}(\boldsymbol{\beta}) = k, \end{aligned} \tag{1}$$

where $\mathbf{Card}(\boldsymbol{\beta}) = k$ is an additional sparsity constraint and $\mathbf{Card}(\cdot)$ counts the number of nonzero components. The problem is non-convex, combinatorial and NP-hard. Since, least square problem can be viewed as Generalized Rayleigh Quotient problem [14], an efficient greedy approach similar to the one proposed in [15] can be adopted here. In other words, the optimal solution to sparse generalized Eigen-value decomposition [15] is also the optimal solution to our sparse least square regression.

To improve the generalization performance, a simple decision stump is introduced to each rectangular feature. Hence, each feature value is represented by a decision stump's output (binary response), specifying object or non-object, respectively. The threshold value in threshold function is selected based on AdaBoost sample weights in each iteration.

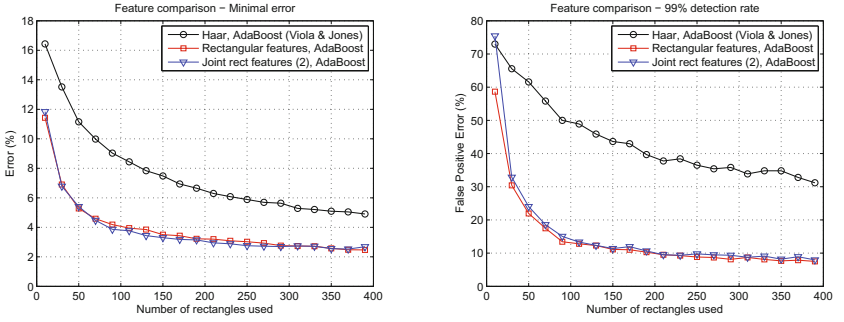
3 Experiments

This section is organized as follows. The data sets used in this experiment, including how the performance is analyzed, are described. Experiments and the parameters used are then discussed. Finally, experimental results and analysis of different techniques are presented.

3.1 Frontal Face Detection

Due to its efficiency, Haar-like rectangle features [1] have become a popular choice as image features in the context of face detection. We compare our rectangular features with Haar-like features. Similar to the work in [1], the weak learning algorithm known as decision stumps is used here due to their simplicity and efficiency.

Performances on Single-node Classifiers. In order to demonstrate the performance of our features, we replace Haar wavelet like features used in [1] with our features. In the first experiment, we compare a single strong classifier learned using AdaBoost with Haar wavelet like features and our proposed rectangular



(a) (b)

Fig. 2. Best viewed in color. Comparison of (a) error rates between Haar-like features and rectangular features. The joint rectangular classifier consists of two rectangles in each weak classifier. (b) false alarm rates on test sets between Haar-like features and rectangular features.

features. The data sets consist of $10,000$ mirrored faces. They were divided into three training sets and two test sets. Each training set contains $2,000$ face examples and $2,000$ non-face examples. The faces were cropped and rescaled to images of size 24×24 pixels. For non-face examples, we randomly selected $5,000$ non-face patches from non-faces images and added $5,000$ difficult non-faces, for a total of $10,000$ patches. For each experiment, three different classifiers are generated, each by selecting two out of the three training sets and the remaining training set for validation. The performance is measured by two different curves:- the test error rate and the classifier learning goal (the false alarm error rate on test sets given that the detection rate on the validation set is fixed at 99%).

Experimental results are shown in Figs. 2a and 2b. The following observations can be made from these curves. Having the same number of learned features, rectangular features achieves lower generalization error rate and false positive error than Haar features. Based on our observations, Haar features seem to perform slightly better than rectangular when the number of features is less than 5. This is not surprising since Haar features contain more variety of shapes than our rectangular features. The first few selected Haar features often combine different parts of the faces and therefore would be more discriminative than our rectangular shape. The performance of our joint rectangular features is also shown in the figure. On face data sets, we observe a lower error rate when we combine two rectangular features. Combining three or more rectangular features does not improve the performance any further.

Performances on Cascades of Strong Classifiers. In the next experiment, we used $2,500$ frontal faces ($5,000$ mirrored faces) that we obtained from [1]. All faces were cropped and rescaled to a size of 24×24 pixels. For non-face examples, we randomly downloaded over $7,000$ images of various sizes from the internet. We used MIT+CMU test sets to test our system. The set contains 130 images

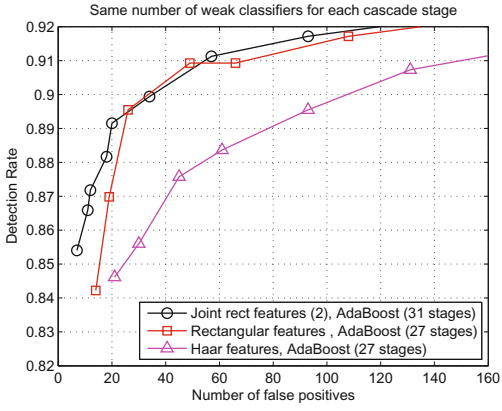


Fig. 3. Performance comparison between our rectangular features, joint rectangular features and Haar features on cascades of strong classifiers

with 507 frontal view faces. We set the scaling factor to 1.2 and window shifting step to 1. The technique used for merging overlapping windows is similar to [1]. Multiple detections of the same face in an image are considered false detections.

For fair evaluation of both rectangular and Haar-like features, we adopted a simple cascade as proposed in [1]. Each cascade layer consists of the same number of features (weak classifiers). The non-face samples used in each cascade layer are collected from false positives of the previous stages of the cascade (bootstrapping). The cascade training algorithm terminates when there are not enough negative samples to bootstrap. Fig. 3 shows a comparison between the Receiver Operating Characteristic (ROC) curves produced by both features. The ROC curves show that rectangular features outperform Haar-like features at all false positive rates. Similar to previous experiment, the combination of two rectangular features in each weak classifier performs best. From the figure, the performance gap between single and joint features is wider at low number of false positives, *i.e.*, at 85% detection rate, joint features achieve 10 less false positives than single features. Experimental results indicate that the type of features we use has a crucial role in the ability of the system to generalize. Fig. 4 shows single and joint rectangular features selected in the first cascade layer. Most selected patches cover the area around the eyes and forehead.

Since, face labeling process is rather tedious and time consuming; it is quite common that the labeled faces are misaligned and rotated. In the next experiment, we compare the performance of rectangular features and Haar-like features on noisy face data sets. In other words, we want to determine how much effect the noisy training data will have on the detection performance. We automatically rotate, shift and illuminate faces in the training sets using some predefined rules. Some of the modified faces are shown in Fig. 5. Similar to previous experiments, we used AdaBoost to train both features. Some readers might point out that AdaBoost is vulnerable to handling noisy data and the use of other classifiers, *e.g.*, LogitBoost [16] and BrownBoost [17], would yield better generalization. However, this would defeat the purpose of comparing Haar fea-

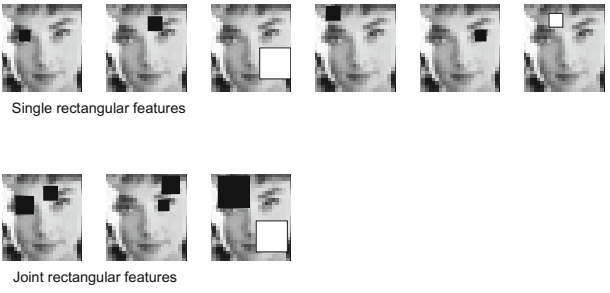


Fig. 4. The first few selected rectangular features from the first layer of cascade. Black boxes indicate HoG features and white boxes indicate LBP feature.

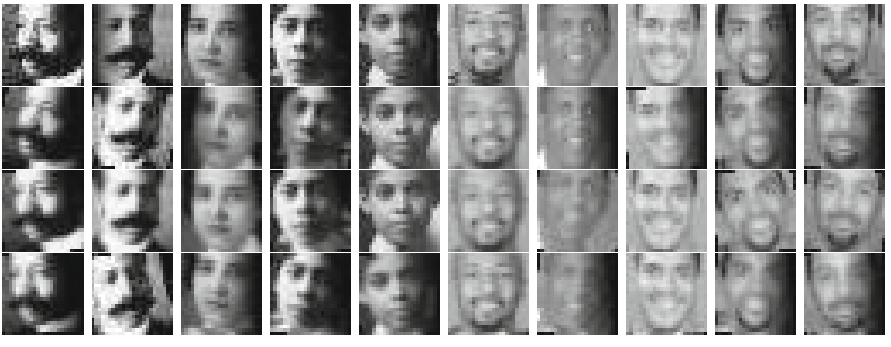


Fig. 5. Manipulated faces. Top row: faces are exposed to random illumination changes and translated randomly. Second row: faces are randomly in plane rotated and exposed to random illumination changes. Third row: faces are randomly in plane rotated and randomly translated. Last row: faces are randomly rotated, randomly translated and exposed to random illumination changes.

Table 1. Performance comparison of classifiers using rectangular features and Haar-like features on noisy data sets. Here, we compare the detection rate on MIT+CMU test sets when the number of false positives is 50. **R** means faces are in plane rotated. **L** means faces are exposed to illumination changes (lighting). **M** means faces are translated by a few pixels (misaligned).

	Rectangle shaped	Haar-like features	Perf. Improvement
Original	0.905	0.878	3.0%
R+L	0.844	0.790	6.3%
M+L	0.856	0.821	4.3%
R+M	0.835	0.814	2.5%
R+M+L	0.826	0.739	12.0%
Average	0.853	0.808	5.5%

tures with rectangular features. Table 1 shows detection rates of both features when trained on different noisy data sets and tested on MIT+CMU test sets. Based on our results, rectangular features are much better at handling noisy training data. We notice less performance drop when the classifier is trained with rectangular features.

The disadvantage of rectangular features compared to Haar-like features is that we now have to keep 8 integral images in the memory for fast feature extraction (signed and unsigned vertical edge responses, signed and unsigned horizontal edge responses and 4 bins for LBP histogram). In terms of an evaluation time, rectangular features have a higher evaluation time than Haar-like features due to an overhead in integral images' calculation.

4 Discussion

In this paper, we proposed a simple and robust local feature descriptor for face detection. Our rectangular features can be denoted by a 4-tuple, (x, y, w, h), where x and y denote the x-coordinate and y-coordinate of the top left position of the block, w and h are the width and height of the rectangles, respectively. Rectangular features are based on simplified HoG and LBP features. Our simplified HoG can be viewed as a sum of edge responses, in vertical and horizontal directions. For unsigned gradients, we apply an absolute value function to edge responses. The absolute value of a real number is its numerical value without its sign. From image processing point of view, the absolute values of the intensity changes represent the magnitude of the edges without taking into consideration the polarity of the edges. Each rectangle is represented by a 4-D feature vector, which is normalized to an ℓ_2 unit length. Simplified binary operator can be viewed as applying the simple threshold function to both vertical and horizontal edge responses. The threshold function can be classified as one form of activation functions commonly used in neural network. The output of the functions takes on the value of 1 or 0 depending on the sign of both horizontal and vertical gradients, (2).

$$\begin{aligned}
\phi_1(x,y) &= \begin{cases} 1 \text{ if } x \geq 0 \text{ and } y \geq 0; \\ 0 \text{ otherwise,} \end{cases} \\
\phi_2(x,y) &= \begin{cases} 1 \text{ if } x \geq 0 \text{ and } y < 0; \\ 0 \text{ otherwise,} \end{cases} \\
\phi_3(x,y) &= \begin{cases} 1 \text{ if } x < 0 \text{ and } y \geq 0; \\ 0 \text{ otherwise,} \end{cases} \\
\phi_4(x,y) &= \begin{cases} 1 \text{ if } x < 0 \text{ and } y < 0; \\ 0 \text{ otherwise.} \end{cases}
\end{aligned} \tag{2}$$

where x and y denote vertical and horizontal edge responses.

An illustration of our rectangular features based on HoG and LBP is shown in Fig. 6. We can generalize rectangular features as follows. First, we apply edge filters to the original image. Edge filter is one of the most popular techniques used

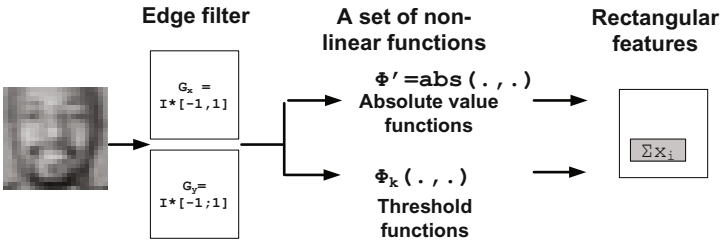


Fig. 6. An illustration of rectangular features

to detect a rate of changes at any given pixel coordinates. Edge responses can be calculated from partial derivatives in horizontal and vertical directions of a given pixel location. After deriving vertical and horizontal edge responses, we apply two non-linear functions to these responses; namely absolute value function and 2-D threshold function. By introducing non-linearity into low-level features, we observe an improvement in the overall performance on visual classification tasks. These non-linear functions might look over-simple. However, many researchers have reported that applying these simple approach often leads to performance improvement in vision applications, *e.g.*, binary operator has been used in LBP to describe image texture as described in Section 2.1, absolute value function has also been used in Speeded Up Robust Features (SURF) [18] where it performs remarkably well in describing key-point descriptor. In summary, our rectangular features consider the change of pixel intensities in a small image neighbourhood to provide an approximate representation of edge responses inside the specific region. This finding raises several open questions related to possible face detection features. In the future we plan to research on learning a more efficient rectangular feature, which would be more memory efficient, work well on general objects and can achieve a comparable speed to Haar-like features.

5 Conclusion

In this work, we proposed the use of simple edge descriptors, which combine the discriminative power of HoG with the strength of LBP operators. Since, single feature is not discriminative enough to separate faces from difficult non-faces, we further improve the generalization performance of our simple features by applying feature co-occurrences. Experimental results show that our new features not only outperform Haar-like features but also yield better generalization when training on noisy data. On average, we achieve a performance improvement of 5.5% when trained with rectangular features.

Acknowledgments. 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. Viola, P., Jones, M.J.: Robust real-time face detection. Int. J. Comp. Vis. 57, 137–154 (2004)
2. Lienhart, R., Maydt, J.: An extended set of haar-like features for rapid object detection. In: Proc. IEEE Int. Conf. Image Process., vol. 1, pp. 900–903 (2002)
3. Li, S.Z., Zhang, Z.: Floatboost learning and statistical face detection. IEEE Trans. Pattern Anal. Mach. Intell. 26, 1112–1123 (2004)
4. Huang, C., Ai, H., Li, Y., Lao, S.: High-performance rotation invariant multiview face detection. IEEE Trans. Pattern Anal. Mach. Intell. 29, 671–686 (2007)
5. Levi, K., Weiss, Y.: Learning object detection from a small number of examples: The importance of good features. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn., Washington, DC, vol. 2 (2004)
6. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn., San Diego, CA, vol. 1, pp. 886–893 (2005)
7. Ojala, T., Pietikinen, M., Menp, T.: Multiresolution gray scale and rotation invariant texture analysis with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell. 24, 971–987 (2002)
8. Wang, X., Han, T.X., Yan, S.: An hog-lbp human detector with partial occlusion handling. In: Proc. IEEE Int. Conf. Comp. Vis. (2009)
9. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comp. Vis. 60, 91–110 (2004)
10. Wu, J., Brubaker, S.C., Mullin, M.D., Rehg, J.M.: Fast asymmetric learning for cascade face detection. IEEE Trans. Pattern Anal. Mach. Intell. 30, 369–382 (2008)
11. Mita, T., Kaneko, T., Stenger, B., Hori, O.: Discriminative feature co-occurrence selection for object detection. IEEE Trans. Pattern Anal. Mach. Intell. 30, 1257–1269 (2008)
12. Avidan, S.: Ensemble tracking. IEEE Trans. Pattern Anal. Mach. Intell. 29, 261–271 (2007)
13. Parag, T., Porikli, F., Elgammal, A.: Boosting adaptive linear weak classifiers for online learning and tracking. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn., Anchorage, pp. 1–8 (2008)
14. Moghaddam, B., Gruber, A., Weiss, Y., Avidan, S.: Sparse regression as a sparse eigenvalue problem. In: Information Theory and Applications Workshop, pp. 219–225 (2008)
15. Moghaddam, B., Weiss, Y., Avidan, S.: Fast pixel/part selection with sparse eigenvectors. In: Proc. IEEE Int. Conf. Comp. Vis., pp. 1–8 (2007)
16. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: a statistical view of boosting. Ann. Statist. 28, 337–407 (2000)
17. Freund, Y.: An adaptive version of the boost by majority algorithm. Mach. Learn. 43, 293–318 (2004)
18. Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: Speeded-up robust features (surf). Comp. Vis. Image Understanding 110, 346–359 (2008)

Multiple Order Graph Matching

Aiping Wang, Sikun Li, and Liang Zeng

School of Computer Science, National University of Defense Technology,
Changsha, Hunan, P.R. China 410073
{ipwang,skli,liangzeng}@nudt.edu.cn

Abstract. This paper addresses the problem of finding correspondences between two sets of features by using multiple order constraints all together. First, we build a high-order supersymmetric tensor, called multiple order tensor, to incorporate the constraints of different orders (e.g., unary, pairwise, third order, etc.). The multiple order tensor naturally merges multi-granularity geometric affinities, thus it presents stronger descriptive power of consistent constraints than the individual order based methods. Second, to achieve the optimal matching, we present an efficient computational approach for the power iteration of the multiple order tensor. It only needs sparse tensor elements and reduces the sampling size of feature tuples, due to the supersymmetry of the multiple order tensor. The experiments on both synthetic and real image data show that our approach improves the matching performance compared to state-of-the-art algorithms.

1 Introduction

Finding correspondences between two sets of features is a very important problem in large research areas of computer vision, such as feature tracking, object recognition, shape matching, wide baseline stereo, 2D and 3D registration. The correspondence problem is normally referred to as graph matching. Given a set of features, a graph is used to represent the important internal structure of the features(e.g., points). The aim of graph matching is to identify a mapping between two node sets of two graphs while keeping as much as possible the constraints between two sets of nodes.

Over the past years, a great number of graph matching approaches were proposed and a comprehensive survey was given by Conte et al. [1]. According to the complexity of geometric relationships, graph matching problems can be divided into two categories. The first class is regarded as solving the linear assignment problem, and the second class is related with solving high-order assignment problems. The linear assignment problem only concerns about the affinity measures between two graph nodes(e.g., the Euclidean distance between two feature points). Such affinity measures rely heavily on local image descriptors (e.g., shape context [2], sift[3]) in many computer vision tasks, so these first-order methods are failure-prone due to the image ambiguities, such as repeated patterns, textures or non-discriminative local appearance. To overcome these problems, the affinity

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 471–482, 2011.

measures defined between pairs of feature correspondences have been proposed to enhance the geometric consistency. If two features in one graph are separately matched to the other two features in another graph, then the relationships derived from these two pairs of matchings are close to each other. The pairwise geometric information of features can help to find correct correspondences even when the features are less or non discriminative.

Recently, several high-order constraints beyond pairwise potentials have been proposed. For example, Zass and Sashua[4] used hypergraphs to describe the high-order affinities, and the problem of searching for high-order feature correspondences is transferred to hypergraph matching. Duchenne et al.[5] introduced a third-order tensor to represent the affinities of feature tuples instead of the affinity matrix, and a high-order power iteration was used to achieve the final matching. Chertok et al.[6] also built a third-order affinity tensor and got the optimal matching by the rank-one approximation of the tensor. However, they derived a marginalization scheme mapping a triplets tensor to a matrix , and the rank-one approximation of the affinity tensor was accomplished by matrix eigendecomposition. In general, high-order graph matching methods, especially tensor based methods, improve the stability of the affinity model and offer much better accuracy than the pairwise based approaches.

With different granularities of geometric constraints, first-order potentials concentrate on individual feature descriptors, while pairwise affinities start to encode geometric properties like local isometry. Moreover, the high-order potentials bring in stronger geometric consistency constraints on larger scales.

In this paper, we present a novel approach to search for feature correspondences by merging multiple order constraints together, which can be called multiple order graph matching. Actually, Leordeanu et al.[7] presented a method to merge information of different orders together, but just limited to second order. Duchenne et al.[5] expanded to high-order case, however, they just combined potentials of different orders in a simple manner, without much consideration of the interrelation among different orders, and their implementation of the tensor power iteration ignored the supersymmetry of the tensor. In general, our contributions are as follows:

1. We build a high-order supersymmetric tensor to accommodate various affinities of different orders. Such multiple order tensor has more powerful expressivity of consistent relationships than the individual order affinity models.
2. We present an efficient approach to implement the power iteration of the multiple order tensor, while just using a small number of its elements. Our method takes full advantage of the supersymmetry of the multiple order tensor, which reduces the sampling size of feature tuples and makes significant savings in computational complexity on power iterations.

The experiments show that our method is more accurate and robust than the state-of-the-art approaches, with competitive computational cost.

The rest of the paper is organized as follows. In Section 2 the formulation of the multiple order tensor is introduced, and the related power iteration method is given in Section3. In Section 4 some implementation issues are discussed, and the experiments are given in Section 5. The last part is the conclusion.

2 Multiple Order Tensor Formulation

Given two sets of features P_1 and P_2, assume that P_1 and P_2 contain N_1 and N_2 features respectively. The correspondences problem is to find an optimal mapping or assignment set C of pairs (i, i'), where $i \in P_1$ and $i' \in P_2$. Each candidate assignment $c_i = (i, i')$ associates with a score or an affinity measure that define how well the feature $i \in P_1$ matches with the feature $i' \in P_2$. Different problems impose different kinds of mapping constraints on C.

In the first order case, the affinity measures M_1 just build on c_i, so the matching cost is defined by an affinity matrix $A \in \mathcal{R}^{N_1 N_2}$ such that $A_{i,i'} = M_1(c_i)$. The optimal assignment is just given by

$$C^* = arg \max_C \sum_C M_1(c_i) \tag{1}$$

subject to the constraint that the matching should be one-to-one.

In the second order case, the pairwise affinities M_2 measure the compatibilities between each pair of assignments c_i and c_j . Therefore, the pairwise affinity matrix $A \in \mathcal{R}^{N_1 N_2 \times N_1 N_2}$ is constituted by $M_2(c_i, c_j)$, and the optimal assignment is achieved by maximizing the sum of corresponding pairwise affinities

$$C^* = arg \max_C \sum_C M_2(c_i, c_j) \tag{2}$$

An important constraint is that A should be non-negative.

The high-order assignment is an extension of the pairwise case. The *nth*-order potentials M_n measure the affinities of n pairs of assignments $(c_{i_1}, c_{i_2}, \ldots, c_{i_n})$. In [5] , the *nth*-order affinities M_n are represented by a *nth*-order tensor $\mathcal{T}_n$, such that $\mathcal{T}_n(i_1, i_2, \ldots, i_n) = M_n(c_{i_1}, c_{i_2}, \ldots, c_{i_n})$. Since $M_n(c_{i_1}, c_{i_2}, \ldots, c_{i_n}) = M_n(\Omega(c_{i_1}, c_{i_2}, \ldots, c_{i_n}))$, where Ω is any permutation of the indices, tensor $\mathcal{T}_n$ is supersymmetric. The final optimal assignment will be estimated by the rank-one approximation of the tensor $\mathcal{T}_n$.

The constraints of different orders vary in granularity of the geometric information. It could improve the expressivity and accuracy of the affinity measurements to use the potentials of different orders in the same time. Leordeanu et al.[7] put unary and pairwise items together into a matrix, while Duchenne et al.[5] combined different orders within one tensor power iteration framework. Here we build a nonnegative supersymmetric tensor to accommodate all the affinities of different orders in a more natural way.

Definition 1 (Multiple Order Tensor). *Given a nonnegative nth-order* ($n \geqslant 3$) $M \times M \times \cdots \times M$ *supersymmetric tensor* $\mathcal{T}_n$*, which satisfies that* $\mathcal{T}_n(l_1, l_2, \ldots, l_n)$

$= \mathcal{T}_n(\Omega(l_1, l_2, \ldots, l_n))$, *where* Ω *is any permutation of the indices. There exist a subset* Ψ *of potential indices* $\{\Phi_1, \Phi_2, \ldots, \Phi_{n-1}\}$, *and a related subset* $\tilde{M}$ *of affinities* $\{M_1, M_2, \ldots, M_{n-1}\}$. $\mathcal{T}_n$ *is a nth-order multiple order tensor when it builds on* $\Psi \cup \{\Phi_n\}$ *and* $\tilde{M} \cup \{M_n\}$, *as*

$$\begin{aligned}
\mathcal{T}_n(\Phi_1(i)) &= M_1(c_i) \\
&\ldots \quad \ldots \\
\mathcal{T}_n(\underset{m<n}{\Phi_m}(i_1, i_2, \ldots, i_m)) &= \underset{\forall p,q \in \{i_1,i_2,\ldots,i_m\}\ p\neq q,\ c_p \neq c_q}{M_m}(c_{i_1}, c_{i_2}, \ldots, c_{i_m}) \\
&\ldots \quad \ldots \\
\mathcal{T}_n(\Phi_n(i_1, i_2, \ldots, i_n)) &= \underset{\forall p,q \in \{i_1,i_2,\ldots,i_n\}\ p\neq q,\ c_p \neq c_q}{M_n}(c_{i_1}, c_{i_2}, \ldots, c_{i_n})
\end{aligned}$$

where the elements M_i *are the ith-order potentials, and the indices are*

$$\begin{aligned}
\Phi_1(i) &= (\underbrace{i, i, \ldots, i}_{n}) \\
&\ldots \quad \ldots \\
\underset{m<n}{\Phi_m}(i_1, i_2, \ldots, i_m) &= \underset{\forall p,q \in \{i_1,i_2,\ldots,i_m\}\ p\neq q,\ c_p \neq c_q}{\Omega}(span(i_1, i_2, \ldots, i_m)) \\
&\ldots \quad \ldots \\
\Phi_n(i_1, i_2, \ldots, i_n) &= \underset{\forall p,q \in \{i_1,i_2,\ldots,i_n\}\ p\neq q,\ c_p \neq c_q}{\Omega}(i_1, i_2, \ldots, i_n)
\end{aligned}$$

where $span(i_1, i_2, \ldots, i_m)$ *stands for all the n-tuples constituted by n elements picked from* $(i_1, i_2, \ldots, i_m)$, *and every member in* $(i_1, i_2, \ldots, i_m)$ *must appear at least once in each one of these n-tuples.*

Take third-order multiple order tensor $\mathcal{T}_3$ for instance. Given three affinities of different orders M_1, M_2, M_3 and the associated indices Φ_1, Φ_2, Φ_3, $\mathcal{T}_3$ can be expressed as

$$\mathcal{T}_3(\Phi_1(i)) = \mathcal{T}_3(i, i, i) = M_1(c_i) \tag{3}$$

$$\begin{aligned}
\mathcal{T}_3(\Phi_2(i, j)) &= \mathcal{T}_3(i, i, j) = \mathcal{T}_3(i, j, i) = \mathcal{T}_3(j, i, i) \\
&= \mathcal{T}_3(i, j, j) = \mathcal{T}_3(j, i, j) = \mathcal{T}_3(j, j, i) = M_2(c_i, c_j)
\end{aligned} \tag{4}$$

$$\begin{aligned}
\mathcal{T}_3(\Phi_3(i, j, k)) &= \mathcal{T}_3(i, j, k) = \mathcal{T}_3(i, k, j) = \mathcal{T}_3(j, i, k) \\
&= \mathcal{T}_3(j, k, i) = \mathcal{T}_3(k, i, j) = \mathcal{T}_3(k, j, i) = M_3(c_i, c_j, c_k)
\end{aligned} \tag{5}$$

According to Def. 1, the *nth*-order multiple order tensor has some different patterns. We do not have to incorporate all the potentials below the *nth* order, while some orders can be omitted, depending on different applications. For example, $\mathcal{T}_4$ may include the first order, the second order and the fourth order, or just the first order and the fourth order, or only the fourth order which becomes the one used in [5].

Definition 2 (Full Order Multiple Order Tensor). *A nth-order multiple order tensor is called full nth-order multiple order tensor only if it includes all the potentials from the first order to the nth order.*

From the definition of the multiple order tensor we can see that the whole tensor with numerous elements can be expressed by small number of elements, which depends on $\#\{\Psi \cup \{\Phi_n\}\}$. Such characteristic makes the computation much simple and efficient, which we will discuss in section 3.

3 Tensor Power Iteration

The graph matching problem is to find an assignment matrix $X \in \{0,1\}^{N_1 \times N_2}$, $\forall i_1 \in P_1, i_2 \in P_2, X_{i_1,i_2} = 1$, when i_1 matches i_2, otherwise $X_{i_1,i_2} = 0$. X can be row-wise vectorized to an assignment vector $\boldsymbol{x} \in \{0,1\}^{N_1 N_2}$. Given a *nth*-order $M \times M \times \cdots \times M$ ($M = N_1 N_2$) dimensions multiple order tensor $\mathcal{T}_n$, the optimal assignment vector $\boldsymbol{x}^*$ can be found by

$$\boldsymbol{x}^* = arg\max_{\boldsymbol{x}} \sum_{i_1,i_2,\ldots,i_n} \mathcal{T}_n(i_1,i_2,\ldots,i_n)\boldsymbol{x}_{i_1}\boldsymbol{x}_{i_2}\cdots\boldsymbol{x}_{i_n} \tag{6}$$

Solving Equ. 6 is known as a *np*-complete problem, while $\boldsymbol{x}^*$ can be estimated by solving the best rank-one approximation $\hat{\mathcal{T}}_n$ of $\mathcal{T}_n$ [8]

$$\hat{\mathcal{T}}_n \overset{def}{=} \lambda\, \boldsymbol{u} \star \boldsymbol{u} \star \cdots \star \boldsymbol{u} = \lambda\, \boldsymbol{u}^{\star n} \tag{7}$$

where λ is a scalar and $\boldsymbol{u} \in \mathcal{R}^{N_1 N_2}$ is an unit-norm vector. $\star$ is called Tucker product[9]. The optimal assignment vector $\boldsymbol{x}^*$ can be derived by discretizing $\boldsymbol{u}$ with many different approaches.

3.1 Symmetric Higher-Order Power Iteration

The higher-order power method(HOPM) is proposed for the rank-one approximation [8], and the version for supersymmetric tensors, called S-HOPM, is given by [9]. The S-HOPM algorithm converges under the assumptions of the convexity for the functional induced by the tensor[9], which are often satisfied in many practical applications. The S-HOPM algorithm is presented as Algorithm 1.

Algorithm 1. Symmetric higher-order power iteration method

Input: *nth*-order supersymmetric tensor $\mathcal{T}_n$
output: unit-norm M-vector $\boldsymbol{u}$
1. Initialize $\boldsymbol{u}_0$, $k = 1$
2. **repeat**
3. $\quad \hat{\boldsymbol{u}}^{(k)} = \mathcal{I} \overset{\mathcal{T}_n}{\star} (\boldsymbol{u}^{(k-1)}) \overset{\mathcal{T}_n}{\star} (n-1)$
4. $\quad \boldsymbol{u}^{(k)} = \hat{\boldsymbol{u}}^{(k)} / \|\hat{\boldsymbol{u}}^{(k)}\|$; $\quad k = k+1$
5. **until** convergence

In Algorithm 1, $\overset{\mathcal{T}_n}{\star}$ is called $\mathcal{T}_n$-product[9], and it satisfies that

$$\hat{\boldsymbol{u}}^{(k)} = \mathcal{I} \overset{\mathcal{T}_n}{\star} (\boldsymbol{u}^{(k-1)}) \overset{\mathcal{T}_n}{\star} (n-1) \Leftrightarrow \tag{8}$$

$$\forall i_1,\ u^{(k)}_{i_1} = \sum_{i_2,\ldots,i_n} \mathcal{T}_{i_1,i_2,\ldots,i_n} u^{(k-1)}_{i_2} u^{(k-1)}_{i_3} \cdots u^{(k-1)}_{i_n} \tag{9}$$

3.2 Symmetric Multiple Order Tensor Power Iteration

For a *nth*-order multiple order tensor $\mathcal{T}_n$, which is supersymmetry and can be expressed by sparse elements, it is convenient to implement Equ. 9 by using elements of different orders respectively. Taking the full third-order multiple order tensor $\mathcal{T}_3$ as an example, according to Def. 2, we have following equations:

$$\forall i \in \Phi_1 \ , \quad u_i^{(k)} = \mathcal{T}_3(i,i,i) u_i^{(k-1)} u_i^{(k-1)} = \mathcal{T}_3(\Phi_1(i)) u_i^{(k-1)} u_i^{(k-1)} \tag{10}$$

$$\begin{aligned} &\forall (i,j) \in \Phi_2 \ , \\ u_i^{(k)} &= \mathcal{T}_3(i,j,j) u_j^{(k-1)} u_j^{(k-1)} + \mathcal{T}_3(i,i,j) u_i^{(k-1)} u_j^{(k-1)} + \mathcal{T}_3(i,j,i) u_j^{(k-1)} u_i^{(k-1)} \\ &= \mathcal{T}_3(\Phi_2(i,j)) (u_j^{(k-1)} u_j^{(k-1)} + 2\, u_i^{(k-1)} u_j^{(k-1)}) \end{aligned} \tag{11}$$

$$\begin{aligned} u_j^{(k)} &= \mathcal{T}_3(j,i,i) u_i^{(k-1)} u_i^{(k-1)} + \mathcal{T}_3(j,i,j) u_i^{(k-1)} u_j^{(k-1)} + \mathcal{T}_3(j,j,i) u_j^{(k-1)} u_i^{(k-1)} \\ &= \mathcal{T}_3(\Phi_2(i,j)) (u_i^{(k-1)} u_i^{(k-1)} + 2\, u_j^{(k-1)} u_i^{(k-1)}) \end{aligned} \tag{12}$$

$$\begin{aligned} &\forall (i,j,l) \in \Phi_3 \ , \\ u_i^{(k)} &= \mathcal{T}_3(i,j,l) u_j^{(k-1)} u_l^{(k-1)} + \quad \mathcal{T}_3(i,l,j) u_l^{(k-1)} u_j^{(k-1)} \\ &= \mathcal{T}_3(\Phi_3(i,j,l))\, 2\, u_j^{(k-1)} u_l^{(k-1)} \end{aligned} \tag{13}$$

$$\begin{aligned} u_j^{(k)} &= \mathcal{T}_3(j,i,l) u_i^{(k-1)} u_l^{(k-1)} + \quad \mathcal{T}_3(j,l,i) u_l^{(k-1)} u_i^{(k-1)} \\ &= \mathcal{T}_3(\Phi_3(i,j,l))\, 2\, u_i^{(k-1)} u_l^{(k-1)} \end{aligned} \tag{14}$$

$$\begin{aligned} u_l^{(k)} &= \mathcal{T}_3(l,i,j) u_i^{(k-1)} u_j^{(k-1)} + \quad \mathcal{T}_3(l,j,i) u_j^{(k-1)} u_i^{(k-1)} \\ &= \mathcal{T}_3(\Phi_3(i,j,l))\, 2\, u_i^{(k-1)} u_j^{(k-1)} \end{aligned} \tag{15}$$

Replacing the Equ. 8 in Algorithm 1 with Eqs. 10 to 15, we extend the general S-HOPM algorithm to the symmetric multiple order tensor power method (Algorithm 2, full third-order multiple order tensor as instance). It is efficient to perform the power iteration by just considering potential items, and the complexity of the whole iteration process only depends on the size of all affinities.

In order to eliminate the overlaps that may appear during the iteration process, we impose a constraint on $\{\Phi_1, \Phi_2, \ldots, \Phi_n\}$ that none of them has repetition inside, which can be expressed as

$$\forall (i_1, i_2, \ldots, i_m), (j_1, j_2, \ldots, j_m) \in \underset{m \leqslant n}{\Phi_m} \ , \ (i_1, i_2, \ldots, i_m) \neq \Omega(j_1, j_2, \ldots, j_m) \tag{16}$$

Such constraint not only keeps the iteration process accurate, but also makes the feature sampling process efficient, which we will discuss in section 4.2.

There are many initialization schemes proposed for the power method [8,9], while here we just use a random scheme to initialize the $\boldsymbol{u}_0$ and make sure it has only positive values. The positive initial vector will make the algorithm converge to a meaningful result.

Algorithm 2. Symmetric multiple order tensor power iteration method

Input: full third-order supersymmetric multiple order tensor $\mathcal{T}_3$
output: unit-norm M-vector $\boldsymbol{u}$

Initialize $\boldsymbol{u}_0$, $k = 1$
repeat
  for all $i \in \Phi_1$
    $u_i^{(k)} = \mathcal{T}_3(\Phi_1(i))u_i^{(k-1)}u_i^{(k-1)}$
  end
  for all $(i,j) \in \Phi_2$
    $u_i^{(k)} = u_i^{(k)} + \mathcal{T}_3(\Phi_2(i,j))(u_j^{(k-1)}u_j^{(k-1)} + 2u_i^{(k-1)}u_j^{(k-1)})$
    $u_j^{(k)} = u_j^{(k)} + \mathcal{T}_3(\Phi_2(i,j))(u_i^{(k-1)}u_i^{(k-1)} + 2u_i^{(k-1)}u_j^{(k-1)})$
  end
  for all $(i,j,l) \in \Phi_3$
    $u_i^{(k)} = u_i^{(k)} + \mathcal{T}_3(\Phi_3(i,j,l))\,2u_j^{(k-1)}u_l^{(k-1)}$
    $u_j^{(k)} = u_j^{(k)} + \mathcal{T}_3(\Phi_3(i,j,l))\,2u_i^{(k-1)}u_l^{(k-1)}$
    $u_l^{(k)} = u_l^{(k)} + \mathcal{T}_3(\Phi_3(i,j,l))\,2u_i^{(k-1)}u_j^{(k-1)}$
  end
  $\boldsymbol{u}^{(k)} = \hat{\boldsymbol{u}}^{(k)}/\|\hat{\boldsymbol{u}}^{(k)}\|$; $k = k+1$
until convergence

The normalization method used in Algorithm 2 is different with the Frobenius norm used in general S-HOPM algorithm. We only perform normalization of each column as the way in [5], and [5] gives a strict proof on convergence of the power iterations for unit norm columns. This method actually makes a tighter relaxation of the assignment matrix $X \in \{0,1\}^{N_1 \times N_2}$, with $\sum_{i_1} X_{i_1,i_2} = 1$, which implies that a feature in the first graph is matched to exactly one in the second graph and the opposite does not hold. Such constraint is reasonable, because two separate graphs may have different number of features to be matched.

Finally, the optimal assignment result is obtained by a natural projection step as choosing the maximum element in each column to build the final discrete assignment matrix.

4 Implementation Issues

4.1 Potential Details

Here we introduce several commonly used affinities from the first order to the third order, which also applied in our experiments, and we design a method to combine all the different potentials within one uniform affinity framework.

First-order potentials. There are many feature descriptors proposed for different kinds of features. We choose shape context[2] of feature point to be the first-order potentials, because the shape context of a point captures the geometric location distribution of the other circumambient points.

Second-order potentials. Pairwise affinities are often defined as

$$M_2(c_i, c_j) = M_2(\{i,i'\},\{j,j'\}) = \exp(-1/\varepsilon^2(\|i-j\|_2 - \|i'-j'\|_2)^2) \qquad (17)$$

where $\varepsilon > 0$ is the is the kernel bandwidth. Such kind of pairwise potentials has a well geometric characteristic called local isometry.

Third-order potentials. In most cases, the triangles formed by three points keep local geometric invariance on scale, rotation and translation. Using the difference of corresponding angles, the third-order affinities are defined as

$$M_3(c_i, c_j, c_k) = M_3(\{i, i'\}, \{j, j'\}, \{k, k'\}) = \exp(-1/\varepsilon^2 \sum\nolimits_{(l,l')} (\|\theta_l - \theta_{l'}\|)^2) \quad (18)$$

where $\varepsilon > 0$ is the is the kernel bandwidth, and the θ_l and θ'_l are the angles of triangles formed by the feature triples (i, j, k) and (i', j', k').

Also there are many other invariant feature potentials of different orders can be used, which depending on different applications.

Uniform potentials measurement. Since potentials of different orders are defined in various manners, we have to bring them into an uniform measurement framework. Given n potentials of different orders $M_1, \ldots, M_n$, the first step is to carry out normalization for each of the potentials $\forall i, M_i = M_i / \|M_i\|_2$. Then we use one affinity function to rebuilt all the potentials, here we use

$$\forall i, M_i = \exp(-1/\varepsilon^2 M_i) \quad (19)$$

where $\varepsilon > 0$ is the is the kernel bandwidth. Different affinity functions can be used here to enhance the discrimination for each of the potentials.

4.2 Sampling

In Section 3.2 it is mentioned that each potential set M_i should have no duplicated items, therefore every feature sampling process for affinities of different orders should make sure that the constraint(formula 16) is satisfied. Actually, such constraint improves the matching accuracy while reduces the sampling size, as will be shown by our experiments.

There exist many different sampling strategies on different purposes. Here we just discuss the general random sampling method. Under the no-repetition constraint, the maximal sampling size is $O(n)$ on a graph with n features in the first order case, and $O(n(n-1)/2)$ in the pairwise case. For higher dth-order potentials, the maximal sampling size is $O((n(n-1)(n-2)\cdots(n-d+1))/d!)$. Apparently, it costs much to use the maximal sampling strategy on both graphs.

For high-order sampling, we adopt a smart sampling scheme as an efficient approximation like the way in [5]. Take third-order potentials as an example, for graph one, only $n_1 * t$ triplets are sampled, and the number of t should just be sure to make every feature in graph one be sampled. For graph two, we use all the possible triangles. Then for each selected triangle of graph one, we find its k nearest neighbors of graph two to build the indices of the tensor. Thus, all the sampling size is $O(n_1 t + n_2(n_2-1)(n_2-2)/3!)$. Fortunately, we find that when two graphs are of the close size, both graphs can use random sampling triplets to build the the indices of the tensor without loss of accuracy, which means the sampling size can be reduced to $O(n_1 t + n_2 t)$.

5 Experiments

The proposed method was applied to artificial graphs as well as real images, and compared to the contemporary state-of-the-art algorithms. For all experiments we used the full third-order multiple order tensor as an example.

5.1 Synthetic Data

We used synthetically generated random graphs to quantify the accuracy and robustness of our method, and made a comparison to the bipartite graph matching method[2](first-order method), the spectral method[10](pairwise method), the third-order tensor based method[5] and its third-order multiple order based method[5]. The task is to recover the point-to-point matching and all results are expressed by the accuracy measured as the proportion of correct matches.

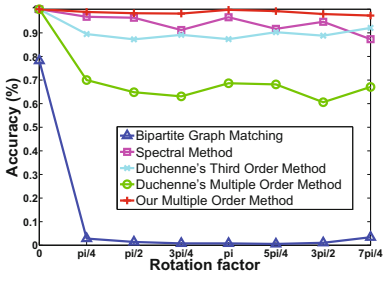


Fig. 1. The result of rotation test

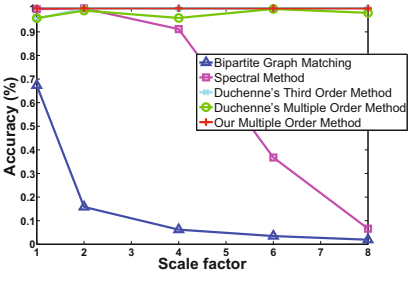


Fig. 2. The result of rescaling test

Four kinds of experiments were carried out: rotation test, rescaling test, distortion test and outlier test. For the first three tests, we generate 50 random points, uniformly distributed in 2D space, as graph P_1 , and graph P_2, with the same number of nodes, is obtained by

rotation test:

$$P_2 = R_\theta \cdot S_\delta \cdot P_1 + N(0, 0.05),\ \theta \in \{0, 7\pi/4\}\ \text{ step by } \pi/4, \forall S_\delta \in (0.5, 1.5)$$

scaling test:

$$P_2 = R_{\delta'} \cdot S \cdot P_1 + N(0, 0.05),\ S \in \{1, 8\}\ \text{ step by } 2, \forall \delta' \in (-10^\circ, 10^\circ)$$

distortion test:

$$P_2 = R_{\delta'} \cdot S_\delta \cdot P_1 + N(0, \sigma),\ \sigma \in [0, 1],\ \forall \delta' \in (-10^\circ, 10^\circ),\ \forall S_\delta \in (0.5, 1.5)$$

where S_δ and $R_{\delta'}$ are little disturbances of scale and rotation, and N stands for Gaussian noise. θ, S and σ are assigned manually. Here rotation means all points in a graph rotate around the straight line perpendicular to the center of the 2D plane with the same angle θ.

For the last outlier test, we generate 20 random points as graph P_1 , and add appointed number of random outliers to graph P_1 to build graph P_2, combining with small random changes on scale and rotation and also with Gaussian noise($\sigma = 0.05$).

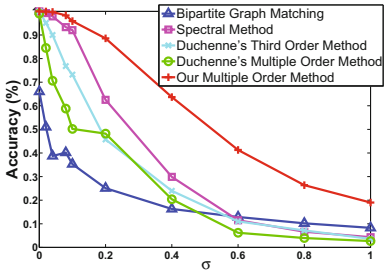


Fig. 3. The result of distortion test

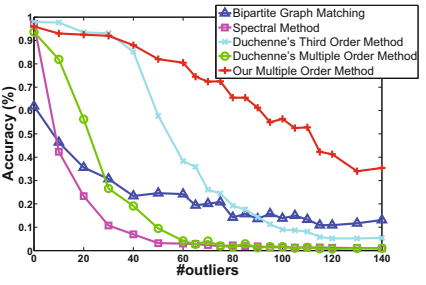


Fig. 4. The result of outlier test

Each kind of experiment was executed 50 times for each different method, with different random graphs generated each time. The mean matching accuracy over all experiments are given in Figure 1 to 4.

From the results of rotation test and rescaling test, we can see that our method remains stable and outperforms others. Under different rotations and scalings, the bipartite graph matching method [2] is easily to fail, while high-order methods are much better due to the rich geometric constraints. However, the multiple order method in [5] does not perform well, because the quantity relationships among elements of different orders are not established accurately. It is also clear that the spectral method [10] cannot deal with large rescaling.

In the distortion test and the outlier test, it is shown that our multiple order method is much more robust than other methods.

In the first three tests(rotation, rescaling and distortion) P_1 and P_2 are of the same size, and we used random sampling scheme on both P_1 and P_2 for third-order sampling. The number of sampling triangles is $50 * t = 50 * 50$ for both P_1 and P_2 in all these three tests. In outlier test, considering the size of P_2 is larger than P_1, we adopted random sampling scheme on P_1 and full sampling scheme on P_2, with $20 * t = 20 * 50$ sampling triangles used by P_1.

For the third-order method [5], we carried out its original sampling strategy, as is random sampling on P_1 and full sampling on P_2. The sampling size is $50 * t = 50 * 100$ on P_1 in the first three tests and $20 * t = 20 * 100$ in the last outlier test. By contrast, out third-order potential sampling size is smaller than [5]. Since the sampling size of the first order and the pairwise are limited, our method has competitive performance on computational cost compared to [5].

5.2 Real Images

We chose images from the Caltech-256 image database to verify our method. Here we find correspondences between two point sets, generated from the silhouettes of two different objects with similar topology structure. First we extracted the main silhouette of each object by edge detector, and then built graphs of points by subsampling those silhouettes. Two graphs were matched to each other by our method, the spectral method [10] and the third-order tensor based method [5] respectively. The results are presented in Figure 5. In order to have a fair comparison, our method and method [5] had same sampling size on the third-order.

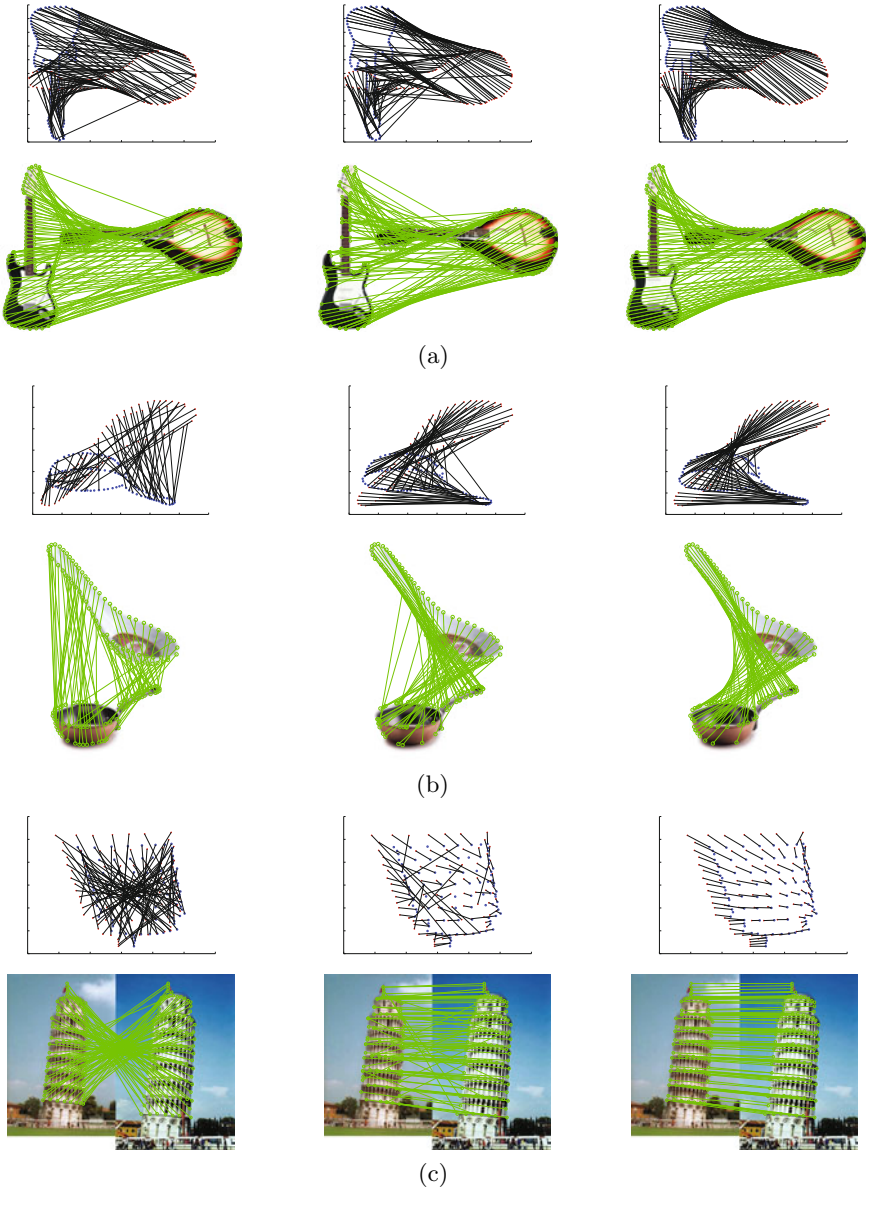


Fig. 5. Results of silhouette matching. The up rows of 5a, 5b and 5c are scatter diagrams. The red × and blue ∘ represent two graphs respectively. The columns of 5a, 5b and 5c, from left to right, are the results of the spectral method[10], the third-order method[5] and our multiple order method.

The spectral method[10] and the third-order tensor based method[5] are prone to error by ambiguities of local context, while because our method merges multi-granularity geometric information, the matching results are much more accurate.

6 Conclusion

In this paper, we presented a multiple order tensor based approach to solve high-order graph matching problems. The multiple order tensor naturally merges multi-granularity geometric affinities of different orders, and enhances the expressivity of consistent constraints compared to the individual order based methods. We also derived an efficient high-order power iteration algorithm for the rank-one approximation of the multiple order tensor. It only needs sparse elements of the tensor, and reduces the sampling size of feature tuples. The experiments on both synthetic and real image data demonstrate that our approach improves the matching performance compared to state-of-the-art algorithms.

Acknowledgement. This work was supported by the National Science Foundation of China(No.90707003, No.60873120, No.60773020, No.60970094), the National High-tech R&D Program of China (No.2009AA01Z301), and the Pre-research funding of National University of Defense Technology (No.JC09-06-01).

References

1. Conte, D., Foggia, P., Sansone, C., Vento, M.: Thirty years of graph matching in pattern recognition. International Journal of Pattern Recognition and Artificial Intelligence 18, 265–298 (2004)
2. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE Trans. Pattern Anal. Mach. Intell. 24, 509–522 (2002)
3. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision 60, 91–110 (2004)
4. Zass, R., Shashua, A.: Probabilistic graph and hypergraph matching. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008), pp. 1–8 (2008)
5. Duchenne, O., Bach, F., Kweon, I., Ponce, J.: A tensor-based algorithm for high-order graph matching. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009), pp. 1980–1987 (2009)
6. Chertok, M., Keller, Y.: Efficient high order matching. IEEE Transactions on Pattern Analysis and Machine Intelligence 99, 32(12), 2205–2215 (2010)
7. Leordeanu, M., Hebert, M.: A spectral technique for correspondence problems using pairwise constraints. In: International Conference of Computer Vision (ICCV 2005), pp. 1482–1489 (2005)
8. Lathauwer, L.D., Moor, B.D., Vandewalle, J.: On the best rank-1 and rank-(r1,r2,.,rn) approximation of higher-order tensors. SIAM J. Matrix Anal. Appl. 21, 1324–1342 (2000)
9. Kofidis, E., Regalia, P.A.: On the best rank-1 approximation of higher-order supersymmetric tensors. SIAM J. Matrix Anal. Appl. 23, 863–884 (2002)
10. Cour, T., Srinivasan, P., Shi, J.: Balanced graph matching. In: Advanced in Neural Information Processing Systems (NIPS 2006) (2006)

Abstraction and Generalization of 3D Structure for Recognition in Large Intra-Class Variation

Gowri Somanath and Chandra Kambhamettu

Video/Image Modeling and Synthesis (VIMS) Lab,
Department of Computer and Information Sciences,
University of Delaware, Newark, DE, USA
http://vims.cis.udel.edu

Abstract. Humans have abstract models for object classes which helps recognize previously unseen instances, despite large intra-class variations. Also objects are grouped into classes based on their purpose. Studies in cognitive science show that humans maintain abstractions and certain specific features from the instances they observe. In this paper, we address the challenging task of creating a system which can learn such canonical models in a uniform manner for different classes. Using just a few examples the system creates a canonical model (COMPAS) per class, which is used to recognize classes with large intra-class variation (chairs, benches, sofas all belong to sitting class). We propose a robust representation and automatic scheme for abstraction and generalization. We quantitatively demonstrate improved recognition and classification accuracy over state-of-art 3D shape matching/classification method and discuss advantages over rule based systems.

1 Introduction

Some specific objects can be described using their image based features, shape and size, however, describing object classes is a much difficult task. Thus a system which learns, abstracts and generalizes autonomously from examples is desirable. The philosophy that object classes must be based on their function/purpose has been known since the early works of Winston [1,2], Brady [3,4] and Minsky [5]. Similar to the idea of Brady et al. in 'The Mechanics Mate' [4], our object classes are based on both purpose and 3D structure of objects. Though much has been theorized and implemented, deployment of such systems has been limited. With growing interest and technology to develop domestic and assistive robots, vision systems capable of object purpose identification are as essential as specific object identification. Our work is motivated and geared towards such applications and hence our choice of object classes include objects used for *Sitting* (chairs, sofa, benches), as *Tables* (coffee tables, dining tables, desks), for *Sleeping* (different types of beds), in *Drinking* (cups, mugs) and as *Bottles*. We ask and answer the following questions, (1) Can all types of objects within a purpose based class be recognized with just a single model? What is the extent of abstraction and generalization possible? (2)How do we represent and learn such a model? Can it be derived automatically and uniformly for all classes? (3)

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 483–496, 2011.

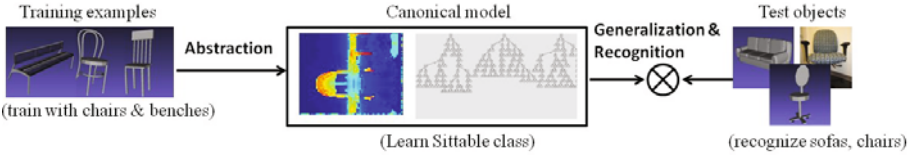


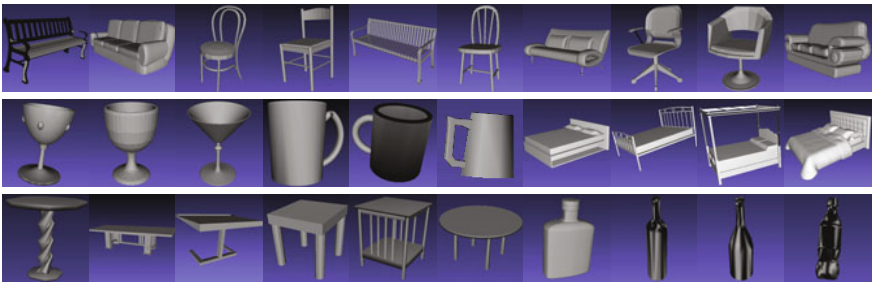
Fig. 1. Given a few training instances, we abstract it to obtain a single canonical model for a class. The canonical model is then used to recognize (previously unseen) members of the class through generalization. The categories have large intra-class variation (for eg., chairs, benches and sofas all belong to the same class).

What are the advantages of this perspective over existing approaches? The main contributions of our work are as follows. We demonstrate that abstraction and generalization can indeed be achieved for classes with very large variation. We propose a representation, COMPAS (Canonical Object Model Based on Purpose And Structure), to derive and represent the abstract model. We present a novel scheme that achieves this in an uniform and automatic fashion. In addition, the proposed scheme has computational advantages such as, training with very few examples per class and need for just one canonical model per class for recognition. The above combined with increased accuracy demonstrates the impact and significance of the novel approach proposed in this paper. The problem of classification is intriguing in both machine vision and human cognitive studies. We model our system based on findings from both areas. It is believed that human classification can be thought of as a multi-system, where both the abstract nature of the object 3D structure/shape and specific details which separate one class from another is used. For example, we would think of a Sittable object as that which has a sufficiently big seat. A chair, sofa, bench can all be described abstractly as containing a seat, a backrest and legs. Even though beds or tables also fulfill the criterion of seating area with support, we are able to separate instances which are previously unseen and can appear significantly different from what has been seen. To obtain a computational model for the above idea, we use spherical functions to represent 3D structures. The canonical model is derived using Gaussian Mixture Models (GMM). The abstraction is achieved by employing a dual component GMM on aligned spherical functions. We show how only a few training examples (order of 5 to 10) are sufficient to learn the canonical model for each class. We define various features and measures on this model, trained using Random Forests, to obtain a robust classifier. We have augmented the Princeton Shape Benchmark dataset, in the relevant classes, to form a large testing dataset. To demonstrate the robustness and practicality of our system, we also test it on a models from stereo images. This shows that our scheme can handle partial and noisy data. We quantitatively compare our method with the state-of-art 3D shape matching/classification scheme implemented on the Princeton 3D Shape Matching engine. This demonstrates that existing shape matching schemes with coarser labels cannot be used to achieve the same goal. We also show that the proposed method is indeed performing recognition at an abstract level, yet maintaining inter-class separation. The rest of the paper is

organized as follows. We discuss in detail the motivation for our scheme and related works in vision and cognitive science in Section 2. The dataset used is detailed in Section 3. The proposed scheme is discussed in Section 4. Results and discussions are presented in Section 5 and we conclude in Section 6.

2 Motivation and Related Work

The problem of classification is intriguing in both machine vision and human cognitive studies. It is interesting to note that some concerns involved in developing computational and algorithmic solutions have also been key inspirations during the formulation of theories about classification tasks in humans. We will now discuss those key issues from the perspective of developing vision systems and findings in cognitive studies, with references to related work in both areas. **Abstraction vs Exemplar:** A long standing debate in both fields has been about the extent of abstraction. The two extreme views have been total abstraction vs. remembering every example/instance of a class. Humans have abstract/conceptual models of chairs – a backrest and seat; tables – flat surface with legs; etc. Thus it may appear that the former theory is more acceptable and that humans just store the central tendency of each class. This view, referred to as prototype view in cognitive literature, was held by many works [6,7,8]. A similar philosophy was used in machine vision by GRUFF based systems [9,10,11]. The GRUFF system used manually created rules/descriptions for each class, for eg: a chair is made of two planar surfaces at approx. right angles etc. However, it was found that this view did not explain how humans handle correlations between different features or large intra-class variability [12,13]. Also, uniform and automatic generation of models is a key requirement for practical vision based systems. At the other end of the spectrum is the exemplar view, which believes that each instance is stored in memory and a one-one match performed during recognition [14,15]. This can be achieved in a vision system by using a 3D shape matching scheme with coarser labels, i.e,storing one or more examples for each class. The class of the best matched example(s) is assigned to the test object. This scheme does not allow for generalization and hence requires that examples are sufficient to span possible intra-class variations. However, it is not always clear how many and what types of examples must be used. As the number of classes increase, **scalability** of such systems for on-board implementation is affected. The same argument of memory economy and variation was also raised in cognitive science [16]. Since the two views were found insufficient in isolation, a multiple-systems perspective was adapted [17,18,19,20,21,22]. The main theme of this intermediate view being that humans 'have multiple categorization utilities that learn different statistical features of the repeating and differentiating environments' [22]. This is the main motivation for our proposed scheme. **Representation:** Minsky in his 'Society of Mind' [5], has stated that the representation plays an important role in the concepts learnt and the generalizations possible. In works like [23,24], appearance and human action are used to identify objects/purpose. Color was used as object features in [23], hence the



Class	Sitting	Drinking	Sleeping	Tables	Bottles
Types	Chair,Bench, Sofa	Cup,Glass, Mug	Bed	Dinning, Coffee	Bottle,Flask
Total(PSB)	87(26)	43(7)	40(8)	73(68)	26(12)

(a) 3D models Dataset

(b) Stereo Dataset

Fig. 2. (a) Some of the objects in the 3D dataset. Objects in the first row all belong to Sitting class, second row contains the Drinking and Sleeping class, third row contains Tables and Bottles. (b)The objects in our stereo dataset and some of the 3D models created of the objects.

system was limited to apply the learnt features to the specific environment and viewpoint. The GRUFF based systems [9,10,11] used superquadrics (planes, sticks, blobs) to represent components. A recognition-by-components (GEONS) [25] style system was then used to describe objects in a class. Segmentation of 3D models into superquadrics is sensitive to noise and hence has been difficult to deploy in stereo or laser based systems. To overcome some of the above limitations, we present methods and representation to capture the abstract structure (3D geometric structure) from just a few training examples. The method is uniform over all classes. We devise features on this representation to capture the specific/differentiating aspects between classes. We would like to stress that no exemplars (3D models) are stored once the initial training has been performed. We demonstrate how the representation can be used to perform generalization by recognition of classes with large intra-class variability.

3 Dataset

We augment the Princeton Shape Benchmark (PSB) [26] and use the extended dataset. The details are provided in the table of Figure 2(a). The number in parenthesis indicates the number of models available from PSB. We obtained

the other models from various sources on the Internet. Some of the models are shown in Figure 2(a). Note however that our classification is different from PSB. Our classes are based on purpose and structure, while it is purely shape on PSB (for eg., the Tables are further separated into round and rectangular in PSB). Same for other classes like Sittable. We have also created a dataset from stereo images. The partial models collected from a single viewpoint are used to test robustness of our scheme. The objects imaged and some of the generated 3D models are shown in Figure 2(b).

4 Creating COMPAS

Intuitively, the abstract/canonical model we wish to derive represents the common structural characteristics of objects in the same class. So our approach is to view each object instance as a combination of canonical part and instance specific variations. Figure 3 shows the overview of the process used to obtain the canonical model from training and using it for recognition. In simple terms, we need to *align* all instances of a class and extract only the *intersection* . This seemingly simple task is challenging due to several factors like intra-class variation, noise in the data and partial data due to single view point or occlusion. Next we seek a suitable representation that can preserve the relevant information from the training examples in a single model (Note that based on the context, model may refer to 3D mesh or the canonical model we build for the class-COMPAS). We can see that some form of a statistical model would be most suitable. We use Gaussian Mixture Models (GMM) to derive the canonical model. We must now determine the domain of the mixture model. We employ a robust representation using spherical extent functions. Features are derived from the canonical model to build classifiers using Random Forests. For recognition similar steps of transformation, alignment and feature extraction is performed on the input model. The learnt classifier is used to determine the membership class of the test object. In the following sub-sections, we provide the intuitive idea and formal algorithms for our scheme.

4.1 Representation

Our representation uses a combination of spherical functions and GMM as described below. Algorithm 1 details the process. Each input 3D model is centered

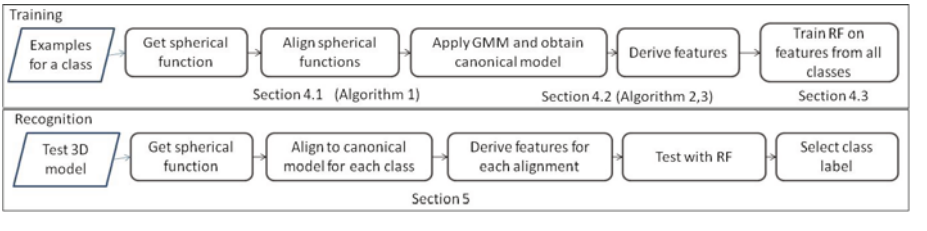


Fig. 3. Overview of the Training and Recognition process

(a) (b) (c) (d) (e) (f)

Fig. 4. Representation: (a) Illustration of the spherical extent function. (b),(c) A chair model and its extent function illustrated as a 2D map. (d),(e) Second chair model and its map. (f) Map of second model (shown in e) aligned with the first (shown in c).

and normalized to fit in a unit cube, i.e, each of the x,y and z co-ordinates lie between 0 and 1 (anisotropic normalization). Then it is represented as a spherical extent function [27] and stored as a 2D map over the two angles. This is derived by intersecting the 3D model with rays originating at the center at discrete angles (θ and ϕ) and storing the length of the ray (r) where it intersects the model. 256 discrete values of the two angles are used in our implementation. The function can be visualized on a 2D map, where the two axes represent the angles and the value is r. In case of models with holes or missing data, r is set to zeros for those rays. This allows using the representation for data obtained from single view, such as stereo. Figure 4(a)-(e) illustrate the spherical extent function and show the 2D map for two chair models. Suppose we were to find *intersection* between the two models using this function, the first step would be to *align* the two models. Finding the best 3D alignment would result in searching the SO(3) space, i.e. the space of all possible three axis rotations. It is known that best 3D alignment results in maximal correlation between the spherical functions. Thus we can perform efficient (fast, low order computations) and robust alignment of the spherical functions in the Fourier domain instead of 3D point cloud alignment. We use the techniques outlined in [28,29] to perform this alignment, which is also robust to missing data and noise. SOFT 2.0 package is used for the alignment. Figure 4(f) shows the spherical function for the second chair after it is aligned to the first.

Initialization: With more such aligned models, the next task is to separate the commonly occurring geometric structures from the distracting designs or structural variations (for e.g, the parts for stylized handles on chairs). The common structures would then constitute a canonical model, specifying the parts which are essential to check for membership in a class. Once the spherical functions have been aligned, the values at a given *pixel* (the value of r at the given θ,ϕ) represent matching structural components of the 3D model. Thus the canonical model can be represented by a similar spherical function where the value at each *pixel* represents the commonly occurring structure over "most" examples. We would like to stress that a simple operation such as mean of all such function maps cannot be performed to obtain the model, as the distribution, in general, is not unimodal. We found using a mixture of two Gaussians to model each *pixel* to give good results. In our implementation, the Gaussian Mixture Model (GMM) is initialized using 5-10 examples for each class. Figure 5 shows an example

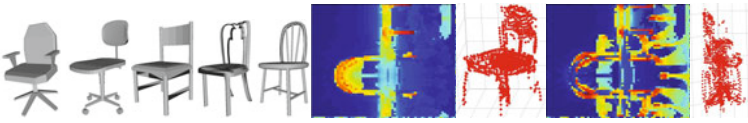
Algorithm 1. Obtaining the Canonical model and map

- Let $M(v, f)$ represent a 3D mesh with vertices v and faces f. We center and normalize $M(v, f)$.
- Let $\Psi(M(v, f)) = S$, where $S(\Theta, \Phi) = r(\Theta, \Phi)$ is the spherical function corresponding to M. $\Theta \in [0, 2\pi], \Phi \in [0, \pi]$.
- Let S_i, S_j be two spherical functions. If we consider $Si = \Lambda(g)Sj$. We can estimate the rotation, $g \in SO(3)$, by correlating the two functions and finding the g_j that maximizes the following integral $\int_{S^2} S_i(\omega)\overline{\Lambda(g)Sj(w)}d\omega$. We denote $S_{ij} = \Lambda(g_j)S_j$.
- With N examples of a class used for initialization, we align the $(N-1)$ 3D models to the first to obtain $S_1, S_{21}, \ldots, S_{N1}$. For each discrete Θ, Φ, we define sets of the form $X^{\Theta,\Phi} = \{S_1(\Theta, \Phi), S_{21}(\Theta, \Phi), \ldots, S_{N1}(\Theta, \Phi)\}$.
- A Mixture Model of K components is defined by the probability density function $p(X) = \sum_{k=1}^{K} p(k)p(X|k)$. Here X denotes the input data points, $p(k) = \pi_k$ is the prior and $p(X|k)$ is the conditional probability density function described by the normal distribution $N(X; \mu_k, \sigma_k)$. The parameters $\{\pi_k, \mu_k, \sigma_k\}$ denote the prior, means and covariance matrix of the components of the GMM. In our implementation we use $K = 2$. The parameters are estimated using Expectation Maximization (EM) algorithm.
- For each Θ, Φ, we obtain a GMM $G^{\Theta,\Phi}$ described by $\{\pi_k^{\Theta,\Phi}, \mu_k^{\Theta,\Phi}, \sigma_k^{\Theta,\Phi}\}$ using $X = X^{\Theta,\Phi}$.
- We form the canonical map C as follows. $C(\Theta, \Phi) = \mu_j^{\Theta,\Phi}$, where $j = argmax_{k=1,2}(\pi_k^{\Theta,\Phi})$ and $\Theta \in [0, 2\pi], \Phi \in [0, \pi]$. Note that C and S are functions defined over the same domain and range. We use this fact during recognition.
- We use the above process to derive the canonical model and map for each of the classes to obtain the set $\{C_c, c \in \{Sitting, Tables, Sleeping, Drinking, Bottles\}\}$.

initialization for chairs using five examples. The mean of the component with highest prior can be visualized as a 2D map, as shown in Figure 5(b). We refer to this 2D map as the canonical map for the class. The value of $r \approx 0$ at a *pixel* indicate the absence/suppression of parts of the 3D structure. Note that here we have shown a model created with only chairs for easier visualization. In the actual experiments, the Sitting class contains examples of chairs, sofa and benches. Figure 5(c) shows a 3D visualization of the map by reconstructing the spherical function and plotting only the tips of the rays. The sparsity of the model is due to sub-sampling. Note that the 3D is shown for illustration purposes only and not used at any processing stage. The mean of the other Gaussian component is shown in Figure 5(d). The algorithm for the above ideas is presented in Algorithm 1. We use the same notation in the following sections.

4.2 Features and Measures

As motivated before, we must devise features and measures to compare both the abstract structure and separate between classes. To enable this, we define two types of features, Type I and Type II. We noted before that the canonical map



(a) (b) (c) (d) (e)

Fig. 5. Initialization: (a) Examples from Sittable class, (b) The canonical map obtained from the examples, and (c) its 3D visualization. (d),(e) The components with lower prior in the GMM. (Only chairs are used here for ease of understanding/visualization. See text).

Algorithm 2. Feature extraction

- Let $\{(S_1, c_1), (S_2, c_2), \ldots, (S_n, c_n)\}$ be the n training examples in spherical function form and known class label.
- For each S_i

  For each class c
  - Align S_i with C_c to obtain S_{ci}
  - Derive Type I features as follows

    $F_1^I(c)$ = correlation between C_c and S_{ci}.

    Let $H(.)$ denote the Spherical Harmonic transform and $P(.)$ the power spectrum. $F_2^I(c) = 1 - |P(H(S_{ci})) - P(H(C_c))|$.
  - Obtain Type II features, $F_1^{II}(c), F_2^{II}(c)$, for aligned S_{ci} using Algorithm 3
  - Obtain feature vector $F_i = [F_1^I \ F_2^I \ F_1^I + F_2^I \ F_1^I - F_2^I \ F_1^I * F_2^I \ F_1^{II} \ F_2^{II}]$

C and the spherical function S are comparable. Thus, during recognition we represent the test 3D model as a spherical function. The extraction of features is described in Algorithms 2 and 3. Type I features measures similarity using correlation of the two functions and L-1 error of the power spectrum difference in the transformed domain. Intuitively, the Type I measures are more sensitive to the dominating structures and capture the gross shape, such as the dominant plane forming the mattress in a bed. Therefore, the need for Type II features which are are sensitive to smaller structures and relative placement of parts. Type II features can be visualized as dividing the 2D map into grids and taking the mean and mode values inside each grid. In our implementation, we use $N = 256, P_{res} = 16$ (see Algorithms 2, 3).

4.3 Classifier

We use Random Forests (RF) [30] to obtain a robust ensemble classifier from the different features. The choice was based on its computational efficiency over other schemes such as neural networks or SVM. Since decision trees mainly learn thresholds to if-else rules, we employ the standard trick to include algebraic combination of matching scores (Type I) as features. Thus for 5 classes, our feature vector F_i is of length 2585 (see Algorithm 2). The feature vectors F_i and the corresponding class labels are used to train the RF. We experimented

Algorithm 3. Obtain Type II features for a spherical function map

- Let $S(\Theta, \Phi)$ be defined for N discrete values of the angles, $\Theta_1, \ldots, \Theta_N$ and $\Phi_1, \ldots, \Phi_N$.
- Choose $P_{res} < N$ and Set $count = 0$.
- for $i = 1 : P_{res} : N$
  for $j = 1 : P_{res} : N$
  - $s = S(\Theta_a, \Phi_b), \forall \Theta_i \leq \Theta_a \leq \Theta_{i+P_{res}}, \Phi_i \leq \Phi_b \leq \Phi_{i+P_{res}}$
  - $count = count + 1$
  - $F_1^{II}(count) = mean(s)$
  - $F_2^{II}(count) = mode(s)$
- Return F_1^{II}, F_2^{II}

with different number of trees and found using more than 400 trees did not improve the overall performance significantly.

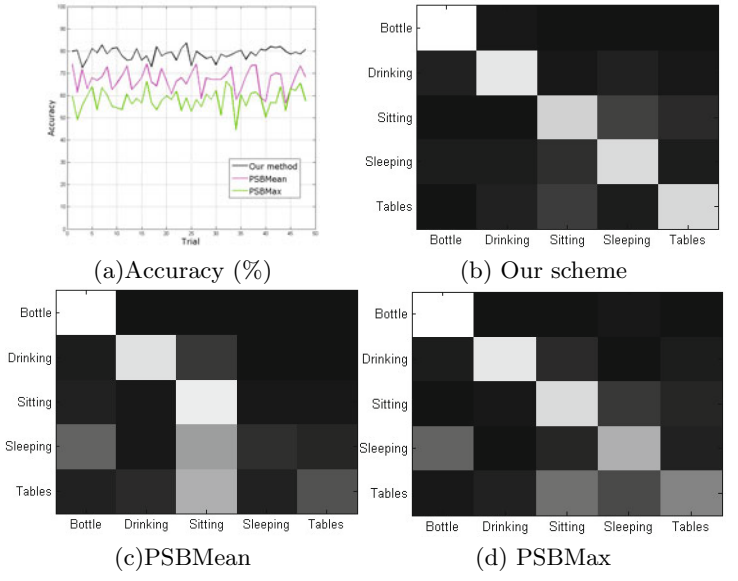
5 Recognition

What is stored for recognition? Once the training has been performed, we can now discard the individual 3D examples used. Only the canonical maps C and the Random Forest structure need to be *stored* for recognition. We use a little over 2 MB to store everything required to perform recognition/classification of 5 classes. Typically a 3D mesh file is of the order of 500KB (0.5MB) to 1MB. Thus with same memory, a maximum of four 3D mesh models can be stored in the exemplar scheme. We would like to stress again that just *one* COMPAS is required to recognize all types of Sitting objects - chairs, benches or sofas. Similarly for the other classes. This would not be possible in the 3D shape matching scheme with just one 3D example.

To use the canonical model and RF for recognition, we follow similar steps of representing the input 3D model as its spherical function. It is then aligned to each of the canonical maps and Type I,II features are derived as before. The feature vector is then tested with the RF to obtain the unknown class label. We compare the performance of our system with the 3D shape matching/classification system on the Princeton 3D model searching engine [31]. We use the implementation provided by the authors at http://www.cs.jhu.edu/ misha/Code/Matching/. Since we use five examples to create the canonical model, we obtain matching score using the author's code for all the five models to be fair. The final score for each class is determined using two methods - mean (PSBMean) and max (PSBMax). The final class is determined by the class with highest score. The recognition experiments are reported below.

5.1 Experiment 1

Our results are geared towards answering the following questions, (1) What is the extent of abstraction achieved? (2) Is the derived canonical model sufficient to recognize class members of varied types? (3) What is the accuracy gain over



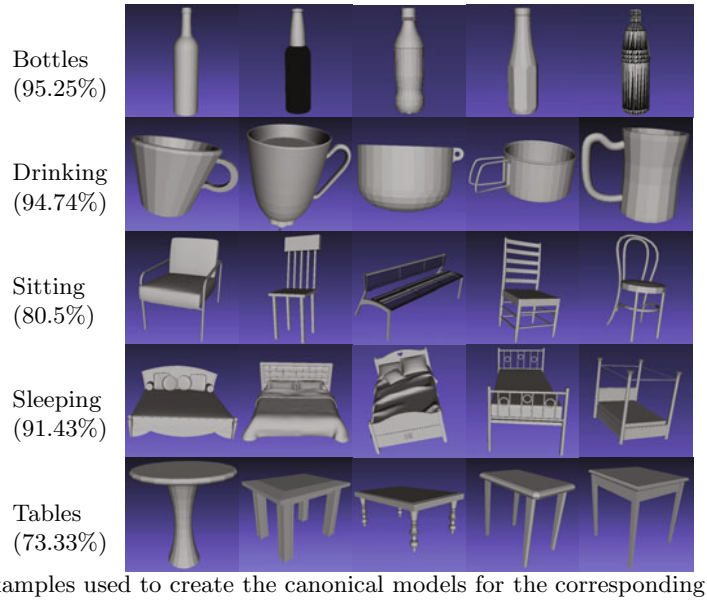
(a)Accuracy (%) (b) Our scheme

(c)PSBMean (d) PSBMax

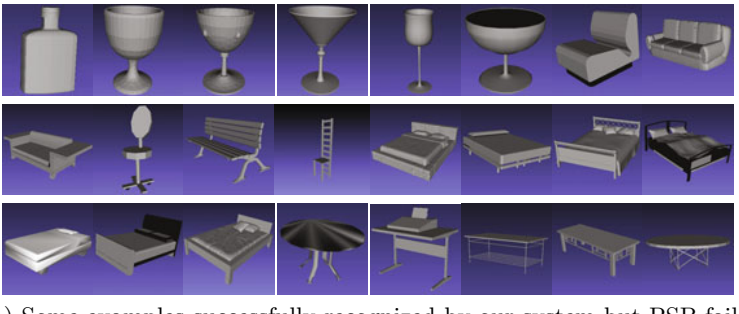
Fig. 6. Experiment 1 results: (a) Overall accuracy of recognition over the different trials (black: Our method, magenta:PSBMean, green:PSBMax), (b) Average confusion matrix for our method, (c) PSBMean and (d) PSBMax.

existing approaches? We report results from 50 trials, with random splits of the dataset. Note that just 5-10 examples per class are used for training in each trial. We first report results on the 3D model dataset described in Section 3. Figure 6 shows the overall accuracy and the average confusion matrix for our scheme and the 3D shape matching/classification algorithm. Our average accuracy is $\approx$ 80% while that of the compared method is $\approx$ 67% (PSBMax) and $\approx$ 58% (PSBMean). The variance in accuracy for our method is $\approx$ 6.7%, while that for PSBMean and PSBMax is $\approx$ 21% and $\approx$ 23%. The confusion matrix shows that though both schemes perform almost equally well in cases like Bottles, where the intra-class variation is comparitively low, our scheme outperforms the 3D shape matching scheme significantly in the Sleeping, Sitting and Tables classes where the intra-class variation is very large. We now look at this in detail.

Extent of abstraction-generalization achieved: It is known that the examples used during training affect the overall performance, especially the extent of abstraction and generalization possible [32, 1]. To show that matching has indeed been done at an abstract level, we look at a case where initialization was performed with examples of a limited type. For instance, consider the case shown in Figure 7. The overall accuracy for the trial was 83.6% (one of the highest), the individual class accuracies is provided in the figure. The corresponding overall accuracy for the PSB based method was 65% (PSBMax) and 59% (PSBMean). For each class, Figure 7(a) shows the models that were used to build the corresponding canonical model. The other results in parts (b)-(d) correspond to the

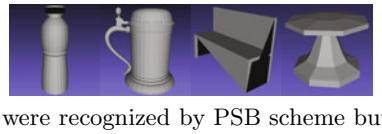


(a) Examples used to create the canonical models for the corresponding class.



(b) Some examples successfully recognized by our system but PSB fails.

(c) Some examples which both schemes failed to recognize.



(d) Examples which were recognized by PSB scheme but not by our scheme.

Fig. 7. Demonstrating extent of abstraction-generalization

trial with the training in (a). Figure 7(b) shows some of the examples which were succesfully recognized by our system but not by the 3D Shape matching (PSB) scheme. Notice that the examples have large variation from those used during the training. For e.g: the bottle (first object in (b)) has a body shape that is quite different from those in the training; similarly though mugs with different handles were used for training the Drinking class, cups and glasses were successfully recognized by our system. This is captured by the Type I features, which helps match parts corresponding to the 'cup' like structure (where the liquid is held). For the case of Sitting, mostly chairs and benches were used for training, but sofas were successfully recognized. The canonical model for Sittable class is dominantly seat and backrest, which allows robust recognition of sofas and different chairs despite other structural variations. Also the anisotropic scaling handles the difference in aspect ratio of parts. The objects in Sleeping class are interesting due to large variation in the distracting elements like pillows, etc. Note that some very simple beds were not recognized by the PSB scheme (first object in last row of (b)). Similarly for the Tables, though only one example had a round top and most had very simple legs, the matched examples have artistic legs or additional items on the surface. Figure 7(c) shows examples which were not recognized by both our scheme and PSB. The first example is that of a bed which was wrongly identified as a table, the second example is of a glass which has two 'cup' like structures. The third and fourth are examples of sofa. The last three are tables. We note that in general, neither scheme can recognize self repeating structures - like the table with multiple shelf like structures.

5.2 Experiment 2

Lastly, we performed experiments on 3D data acquired from stereo. Instead of using volumetric reconstructions, we used the partial models from single viewpoints to test our scheme, since this would be the case for a real world robot. It is possible to use our scheme for recognition of partial data since we define the spherical function to have $r = 0$ in case of models with holes/missing data. Also the alignement scheme is known to be robust to such occlusions. We used the initializations from the previous experiment and performed recognition on the stereo models. We found that our method had an average accuracy of $\approx 65\%$, while PSBMean and PSBMax had an average accuracy of $\approx 51\%$ and $\approx 56\%$, respectively.

6 Future Direction and Conclusion

The results from Experiment 1 showed that our scheme was indeed able to capture abstract structures and perform generalization for successful recognition in classes with large intra-class variation. We also saw that this did not affect the inter-class separation significantly. In future, we seek to perform detailed experiments to derive heuristics which will guide us during training. On one hand, it may seem that providing examples with large variance during training

would lead to best overall performance. But the case shown in Experiment 1 does not fit this trend. We may argue that this is due to lack of consensus on the canonical structure. The question about the characteristics of the training samples, and their effect on the balance between individual class performance and inter-class separation, remains unanswered in many fields related to learning. Since our system is the first to allow such abstraction-generalization in a uniform and automatic way over many classes, it would now be possible to search for heuristics of the above nature for such vision systems.

In this paper, we presented a novel scheme for object classification and recognition. Our contributions are as follows (1) We proposed a novel representation for the canonical model, suitable for extracting features for the set goals. We showed the robustness of the scheme even on partial and noisy real world data acquired from stereo cameras. (2) We provided results to demonstrate that abstraction and generalization was indeed achieved, allowing recognition despite large intra-class variations (for e.g: recognizing sofas when chairs, benches dominated the training set). (3) We have proposed a novel automatic scheme which takes the multi-system approach to classification, motivated from findings about human classification task. (4) We have demonstrated that the system is more robust, scalable and practical compared to existing approaches. (5) We have also shown how 3D shape matching schemes with coarser class labels cannot be employed for the task. We quantitatively demonstrated increase in recognition performance and memory efficiency.

References

1. Winston, P.H.: Learning structural descriptions from examples. In: Winston, P.h. (ed.) The Psychology of Computer Vision, McGraw-Hill, New York (1975)
2. Winston, P., Binford, T., Katz, B., Lowry, M.: Learning physical descriptions from functional definitions, examples,and precedents. In: Proc. Int. Symp. Robotics Research, vol. 1. MIT Press, Cambridge (1984)
3. Connell, J.H., Brady, M.: Generating and generalizing models of visual objects. Artif. Intell. 31, 159–183 (1987)
4. Brady, M., Agre, P.E., Braunegg, D.J., Connell, J.H.: The mechanics mate. In: Advances in Artificial Intelligence, pp. 79–94 (1985)
5. Minsky, M.: The society of mind, pp. 79–94 (1985)
6. Posner, M.I., Keele, S.W.: On the genesis of abstract ideas. J. Exp. Psychol. 77, 353–363 (1968)
7. Reed, S.K.: Pattern recognition and categorization. Cognitive Psychology 3, 382–407 (1972)
8. Smith, J.D., Paul, J.: Distinguishing prototype-based and exemplar-based processes in dot-pattern category learning. Processes in Category Learning, Journal of Experimental Psychology: Learning, Memory, and Cognition, 800–811 (2002)
9. Stark, L., Bowyer, K.: Achieving generalized object recognition through reasoning about association of function to structure. IEEE Transactions on Pattern Analysis and Machine Intelligence 13, 1097–1104 (1991)
10. Stark, L., Bowyer, K.: Function-based generic recognition for multiple object categories. CVGIP: Image Understanding 59, 1–21 (1994)

11. Pechuk, M., Soldea, O., Rivlin, E.: Learning function-based object classification from 3d imagery. Computer Vision and Image Understanding 110, 173–191 (2008)
12. Fried, L.S., Holyoak, K.J.: Induction of category distributions: A framework for classification learning. Journal of Experimental Psychology: Learning, Memory, and Cognition 10, 234–257 (1984)
13. Rips, L.J.: Similarity, typicality, and categorization, pp. 21–59 (1989)
14. Estes, W.K.: Array models for category learning. Cognitive Psychology, 500–549 (1986)
15. Brooks, L.R.: Nonanalytic concept formation and memory for instances. In: Rosch, E., Lloyd, B.B. (eds.) Cognition and Categorization, pp. 169–211 (1978)
16. Rosch, E.: Principles of categorization. In: Rosch, E., Lloyd, B.B. (eds.) Cognition and Categorization, pp. 27–48 (1978)
17. Ashby, F.G., Alfonso-Reese, L.A., Turken, A.U., Waldron, E.M.: A neuropsychological theory of multiple systems in category learning. Psychological Review, 442–481 (1998)
18. Ashby, F.G., Ell, S.W.: The neurobiology of human category learning. Trends in Cognitive Science, 204–210 (2001)
19. Erickson, M.A., Kruschke, J.K.: Rules and exemplars in category learning. Journal of Experimental Psychology: General, 107–140 (1998)
20. Love, B.C., Medin, D.L., Gureckis, T.M.: A network model of category learning. Psychological Review, 309–332 (2004)
21. Vanpaemel, W., Storms, G.: In search of abstraction: the varying abstraction model of categorization. Psychonomic Bulletin and Review, 732–749 (2008)
22. Smith, J.D., Chapman, W.P., Redford, J.S.: Stages of category learning in monkeys (macaca mulatta) and humans (homo sapiens). Journal of Experimental Psychology: Animal Behavior Processes, 39–53 (2010)
23. Veloso, M.M., Rybski, P.E., von Hundelshausen, F.: Focus: a generalized method for object discovery for robots that observe and interact with humans. In: HRI 2006: Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-Robot Interaction, pp. 102–109. ACM, New York (2006)
24. Gupta, A., Kembhavi, A., Davis, L.S.: Observing human-object interactions: Using spatial and functional compatibility for recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 31, 1775–1789 (2009)
25. Biederman, I.: Recognition-by-components: A theory of human image understanding. Psychological Review 94, 115–147 (1987)
26. Shilane, P., Min, P., Kazhdan, M., Funkhouser, T.: The princeton shape benchmark. Shape Modeling International (June 2004)
27. Saupe, D., Vranic, D.V.: 3D model retrieval with spherical harmonics and moments. In: Radig, B., Florczyk, S. (eds.) DAGM 2001. LNCS, vol. 2191, pp. 392–397. Springer, Heidelberg (2001)
28. Kostelec, P.J., Rockmore, D.N.: Ffts on the rotation group. Technical report (2003)
29. Kostelec, P., Rockmore, D.: Ffts on the rotation group. Journal of Fourier Analysis and Applications 14, 145–179 (2008)
30. Breiman, L.: Random forests. Machine Learning, 5–32 (2001)
31. Princeton: 3d model search engine, http://shape.cs.princeton.edu/search.html
32. Winston, P.H.: Learning structural descriptions from examples (1970)

Exploiting Self-similarities for Single Frame Super-Resolution

Chih-Yuan Yang, Jia-Bin Huang, and Ming-Hsuan Yang

Electrical Engineering and Computer Science
University of California at Merced
Merced, CA 95343, USA

Abstract. We propose a super-resolution method that exploits self-similarities and group structural information of image patches using only one single input frame. The super-resolution problem is posed as learning the mapping between pairs of low-resolution and high-resolution image patches. Instead of relying on an extrinsic set of training images as often required in example-based super-resolution algorithms, we employ a method that generates image pairs directly from the image pyramid of one single frame. The generated patch pairs are clustered for training a dictionary by enforcing group sparsity constraints underlying the image patches. Super-resolution images are then constructed using the learned dictionary. Experimental results show the proposed method is able to achieve the state-of-the-art performance.

1 Introduction

Super-resolution algorithms aim to construct a high-resolution image from one or multiple low-resolution input frames [1]. They address an important problem with numerous applications. However, this problem is ill-posed because the ground truth is never known, and numerous algorithms are proposed with different assumptions of prior knowledge so that extra information can be exploited for generating high-resolution images from low-resolution ones. Existing super-resolution algorithms can be broadly categorized into three classes: reconstruction-based, interpolation-based, and example-based approaches.

Interpolation-based super-resolution methods assume that images are spatially smooth and can be adequately approximated by polynomials such as bilinear, bicubic or level-set functions [1,2,3]. This assumption is usually inaccurate for natural images and thus over-smoothed edges as well as visual artifacts often exist in the reconstructed high-resolution images. These edge statistics can be learned from a generic dataset or tailored for a particular type of scenes. With the learned prior edge statistics, sharp-edged images can be reconstructed well at the expense of losing some fine textural details.

For reconstruction-based algorithms, super-resolution is cast as an inverse problem of recovering the original high-resolution image by fusing multiple low-resolution images, based on certain assumed prior knowledge of an observation model that maps the high-resolution image to the low resolution images [4, 5].

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 497–510, 2011.

Each low-resolution image imposes a set of linear constraints on the unknown high-resolution pixel values. When a sufficient number of low-resolution images are available, the inverse problem becomes over-determined and can be solved to recover the high-resolution image. However, it has been shown that the reconstruction-based approaches are numerically limited to a scaling factor of two [5].

For example-based methods, the mapping between low-resolution and high-resolution image patches is learned from a representative set of image pairs, and then the learned mapping is applied to super resolution. The underlying assumption is that the missing high-resolution details can be learned and inferred from the low-resolution image and a representative training set. Numerous methods have been proposed for learning the mapping between low-resolution and high-resolution image pairs [3,6,7,8,9,10,11] with demonstrated promising results.

The success of example-based super-resolution methods hinge on two major factors: collecting a large and representative database of low-resolution and high-resolution image pairs, and learning their mapping. Example-based super-resolution methods often entail the need of a large dataset to encompass as much image variation as possible [3,6,7,8,9,10,11] with ensuing computational load in the learning process. Moreover, the mapping learned from a general database may not be able to recover the true missing high-frequency details from the low-resolution image if the input frame contains textures that do not appear in the database. For example, the mapping function learned from low-resolution/high-resolution image pairs containing man-made objects (e.g., buildings or cars) is expected to perform poorly on natural scenes. Furthermore, the rich image structural information contained in an image is not exploited. In light of this, Glasner et al. [12] propose a method that exploits patch redundancy among in-scale and cross-scale images in an image pyramid to enforce constraints for reconstructing the unknown high-resolution image.

In [10], Yang et al. present a super-resolution algorithm by employing sparse dictionary learning on high-resolution and low-resolution images. In this algorithm, the low-resolution images are considered as a downsampled version of high-resolution ones with the same sparse codes. Using a representative set of image patches, a dictionary (or bases) is learned for sparse coding using both high-resolution and low-resolution images. Their approach performs well under the assumption that image patches of the input image are similar to the ones in the training data, e.g., similar types of images. Existing dictionary learning algorithms often operate on individual data samples without taking their self-similarity into account in searching for the sparsest solutions [13]. Observing this, Mairal et al. [14] recently propose an algorithm exploiting the intuition that similar patches in an image should admit similar sparse representation over the dictionary. By enforcing group sparsity, their experimental results on image denoising and demosaicing demonstrate improvements over existing methods.

We propose a super-resolution method that exploits self-similarities and group structural constraints of image patches using only one single input frame. In contrast to [10], our algorithm exploits patch self-similarity within the image and introduces the group sparsity for better regularization in the reconstruction

process. Compared with [14], we exploit not only the patch similarity within scale but also across scales. In addition, we are the first to show structural sparsity can be successfully applied to the image super-resolution (which is not a trivial extension). Different from [12], we enforce constraints in constructing high-resolution image patches within an image pyramid, and exploit group sparsity and generate better super resolution images. Experimental results show the proposed method is able to achieve the state-of-the-art performance for image super resolution using one single frame.

2 Proposed Algorithm

We present the proposed algorithm in this section. Our approach exploits both patch similarity across scale and group structural constraint underlying the natural images. In contrast to existing super-resolution algorithms that resort to a large data of disparate images, we show that the training patches generated directly from the input image itself facilitate finding more similar patches.

Our algorithm consists of two main steps in which we exploit self-similarities among image patches. We first generate high-resolution/low-resolution patch pairs from one single frame by exploiting self-similarities. To generate high-resolution/low-resolution patch pairs from one single frame, we create an image pyramid and build the patch pairs between corresponding high-resolution/low-resolution images. As shown in [12], the use of an image pyramid provides an effective method to generate a sufficient number of high-resolution patches from low-resolution ones.

After creating high-resolution/low-resolution patch pairs, we enforce the group sparsity constraints among similar patch pairs. The group sparsity constraints have been shown to be effective for image denoising and demosaicing [14]. In contrast to [14], we exploit not only the patch similarity within image scale but also across image scale. In addition, we show that structural sparsity can be successfully applied to the image super-resolution. We present the details of our algorithm in the following sections.

2.1 Exploiting Self-similarities to Generate Example Pairs

In the first step, we generate a set of high-resolution/low-resolution patch pairs from one single input image. These generated patch pairs are used to construct the output high-resolution image in the second step. Conventionally, the source of image pairs for example-based algorithms can be extracted from an extrinsic large dataset that encompasses a wide range of scenes or a category-specific one (e.g., [6,10]). Alternatively, such image pairs can be extracted intrinsically from one single frame (e.g., [12]). The advantage of using extrinsic dataset is the availability of plentiful patch pairs, which may facilitate finding matches between high-resolution and low-resolution image patches. However, the drawback is the ensuing problem with large image variation inherent among image

pairs from diverse sources. Consequently these algorithms may find similar low-resolution patches from the dataset, but the paired high-resolution patches are not necessarily suitable for constructing high quality super-resolution images.

To avoid this problem, we generate patch pairs naturally bearing strong similarities directly from the input low-resolution image itself. Motivated by the observations of [12], we build image patch pairs from an image pyramid to provide highly similar patch pairs.

Assume the relationship between high-resolution image I_h, and low-resolution image I_l is

$$I_l = (I_h * B) \downarrow_s, \tag{1}$$

where $*$ is a convolution operator, B is an isotropic Gaussian kernel, and $\downarrow_s$ is a subsampling operator with scaling factor s. From an input image I_0 shown in Fig. 1, we first generate low-resolution images I_k $(k = -1, \ldots, -n)$. By well controlling the scaling factors and the variance parameters of the Gaussian kernels, it is possible to create high-resolution patches by exploiting self-similarity among the input image and generated low-resolution images. Fig. 1 illustrates the concept, and Proposition 1 states the relationship between scaling factors and the corresponding Gaussian variance parameters.

Proposition 1. *For any two downsampled images* $I_p = (I_0 * B_p) \downarrow_{s_p}$ *and* $I_q = (I_0 * B_q) \downarrow_{s_q}$ *of the image pyramid, the variances of their Gaussian kernels are related by* $\sigma_p^2 = \sigma_q^2 \cdot \log(s_p)/\log(s_q)$.

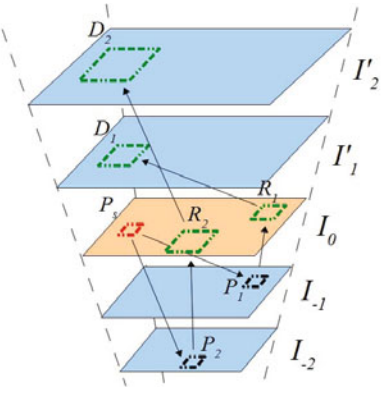


Fig. 1. Exploiting cross-scale patch redundancy in an image pyramid: I_0 is the input image. I_{-1} and I_{-2} are downsampled layers from I_0. The pixels of I'_1 and I'_2 are copied and enlarged from image patches of I_0. For a source patch P_s in I_0, several similar patches (P_1 and P_2) can be found in lower-resolution images (I_{-1} or I_{-2}). For each found patch (P_1 or P_2), a corresponding region (R_1 or R_2) in I_0 are determined. Similarly, a corresponding region (D_1 or D_2) are determined by two factors: (1) the region of source patch P_s, (2) the layer index of the found patch (-1 of I_{-1} or -2 of I_{-2}). Finally, the intensity value of R_1 are copied to D_1 with enlarged area, so as R_2 to D_2.

The proof of this proposition is presented in Appendix 1. We assume the input image I_0 is a downsampled result from an unknown high-resolution image I_k ($k \geq 1$), so that we can exploit patch similarity across scales to fill regions in I_k. We set $s_k = s^{k/n}$ ($k = -1, \ldots, -n$) where s is the expected scaling factor for final output image and n is the number of low-resolution images. This exponential setting is critical because our goal is to create high-resolution/low-resolution patch pairs for second part. Only with this setting described in Proposition 1, the Gaussian kernel variances between I_k to I_{k-n} are the same as I_n to I_0.

For a source patch P_s in the input image I_0, we use the approximate nearest neighbor algorithm [15] to find most similar patches in low-resolution images. Assume two patches are found, i.e., P_1 and P_2 in Fig. 1, their corresponding regions (R_1 and R_2) in I_0 have larger size than P_1 and P_2. Similarly any image patch P_s of I_0 can be assumed to be generated by high-resolution images with Equation 1, and the corresponding regions in the high-resolution images are D_1 and D_2. The relationship between P_k to R_k should be similar as P_s to D_k, and thus we set D_k to have the same intensity as R_k. However, P_s is not completely the same as P_k and R_k is not completely the same as D_k. We compute their weights based on their similarity with $\exp(-\|P_s - P_k\|^2/\sigma^2)$ to average the overlapped high-resolution patches, where σ controls the degree of similarity.

Denote the high-resolution images are I'_1 and I'_2 in Fig. 1, they contain many copied patches but may have some uncovered regions (i.e., some source patches in I_0 may not find similar patches in the image pyramid). We fill the uncovered area with the back projection algorithm [4] for improving image resolution. Because the blur kernels are known in our formulation, we generate high-resolution images by compensating low-resolution images

$$I_h = I''_h - (I''_l - I'_l) \uparrow_s, \tag{2}$$

where I''_h is an initial high-resolution image, I''_l is the image generated by I''_h in Equation 1, and I'_l is the images where D_k is copied to. The upsampling operator $\uparrow_s$ we use here is bicubic interpolation. If I'_l has uncovered areas, we ignore these regions and set their pixel values to to zero. We generate the initial I''_n with bicubic interpolation of I_0, and compensate I'_n to I_0. We summarize the first step to generate high-resolution/low-resolution image pairs in Algorithm 1.

2.2 Exploiting Group Self-similarities to Construct High-Resolution Images

The method presented in Section 2.1 can generate a high-resolution image H, but the resulting image may contain significant amount of noise. In this section we propose a method to further refine it by exploiting the group sparsity constraints among image patches. As the high-resolution image H and low-resolution image L are known, and the width of the Gaussian kernel σ is also known, we can generate several high-resolution images from H by the downsampling process described in Equation 1.

From the first step, we have $n + 1$ pairs of images between I_k and I_{k-n} ($k = 0, \ldots, n$). We form image pairs that every low-resolution patch in I_{k-n} has

Algorithm 1. Construct high-resolution images from single input frame

Data: Input image L, Zooming factor z, Gaussian kernel variance σ_6^2, Number of similar patches m, Similarity weight parameter σ_w, Back-projection loop number l_b

Result: High-resolution images I_1 to I_n (n is decided by z)

Set $I_0 = L$ with resolution (h_0, w_0);
for $k = 1, \ldots, 6$ **do**
    Set scaling factor $s_{-k} = (1/1.25)^k$;
    Compute convoluted image C_{-k} by convolving I_0 with a Gaussian kernel whose variance $\sigma_{-k}^2 = \sigma_6^2 * \log(k)/\log(6)$;
    Set $h_{-k} = h_0 * s_{-k}$, and $w_{-k} = w_0 * s_{-k}$ (possibly non-integer);
    Compute image I_{-k} by subsampling C_{-k} to the resolution (h_{-k},w_{-k});
for $k = 0, \ldots, 5$ **do**
    for each 5×5 patch P_s in I_{-k} **do**
        Compute the corresponding region R_s in $C_{-(k+1)}$ (boundary coordinates of R_s are usually non-integer);
        Compute Q_s by subsampling R_s into a 4×4 patch;
        Save patch pair (Q_s, P_s) into patch pair database B;
Compute number of upsampling image n=roundup($\log(z)/\log(1.25)$);
for $k = 1, \ldots, n$ **do**
    Compute image I_k's resolution as $(h_0 \times (1.25)^k, w_0 \times (1.25)^k)$;
    for each 5×5 region in I_k **do**
        Compute the corresponding region R_q in I_{k-1} (boundary coordinates of R_q are usually non-integer);
        Compute query patch Q_q by subsampling R_q into a 4×4 patch;
        Query Q_q in database B to find similar patches $Q_1 \sim Q_m$ with paired 5×5 patches $P_1 \sim P_m$ and difference value $d_t = \|Q_q - Q_t\|_2$;
        for $t = 1, \ldots, m$ **do**
            Compute patch weight $w_m = \exp(-d_t/\sigma_w)$;
            Record each patch P and weight w;
    Compute average image A by weighted average overlapped patches $\{P\}$ and weights $\{w\}$;
    Set scaling factor $s_k = 1.25^k$;
    Compute Gaussian kernel whose variance $\sigma_k^2 = \sigma_6^2 * \log(k)/\log(6)$;
    Set the initial value of back-projected image Y as A;
    for $t = 1, \ldots, l_b$ **do**
        Compute back-projected image Y respect to I_0 with Gaussian projection kernel (variance = σ_k^2), downscale and upscale factor s_k, back-projection kernel the same as projection kernel;
    Set $I_k = Y$;
    Add patch pairs (Q,P) to B from image pairs I_{k-1} and I_k as above;

Algorithm 2. Refine image through group sparse coding

Data: Image Pyramid $\{I_k\}$ $k = -6, \ldots,$ n , Zooming factor z, Gaussian kernel variance σ_6^2, Low-resolution patch size m, Cluster number c, Group sparsity threshold δ, Dictionary size d, Dictionary update loop number K

Result: Refined high-resolution image H

for *k=0, . . . , 6* **do**
- Denote low-resolution image $L_k = I_{-k}$;
- Compute expected scaling factor $s = 1.25^{-k} * z$ and index t =roundup($\log(s)/\log(1.25)$);
- Denote upsampled image $I_s = I_t$;
- Set $\sigma^2 = \sigma_6^2 * 6 * \log(1.25)/\log(s)$;
- Compute I_c by convolving I_s with a Gaussian kernel whose variance is σ^2;
- Set expected resolution $(h_h, w_h) = (s * h_0, s * w_0)$ where (h_0, w_0) is I_0's resolution;
- Compute H_k by subsampling I_c to resolution (h_h, w_h) ;
- **for** each $m \times m$ patch P_i^l on L_k **do**
  - Set patch P_i^h = the corresponding $mz \times mz$ patch of P_i^l on H_k;
  - Compute high-resolution feature vector $f_i^{h,r} = P_i^h - \text{mean}(P_i^h)$;
  - Compute low-resolution feature vector $f_i^{l,r}$ with gradient vectors P_i^l ;
  - Normalize feature vector $f_i^{h,r}$ to $f_i^{h,n}$ and record the norm value v_i^h;
  - Normalize feature vector $f_i^{l,r}$ to $f_i^{l,n}$;
  - Concatenate vectors $f_i^{h,n}$ and $f_i^{l,n}$ to single vector f_i^c;
  - Normalize vector f_i^c to vector y_i, and save f_i^c's norm value v_i^c;
- Cluster all $\{f_i^{l,r}\}$ by K-means clustering to get c clustering sets $\{U_j\}$, $j = 1 \ldots c$, from vector set. Each U_j contains several indexes of similar $f^{l,r}$;
- Denote Y as all vectors $\{y_i\}$ and set initial dictionary D^0 = first d non-repeated y_i vectors;
- **for** *k=1 , . . . , K* **do**
  - For every cluster U_j, find the coefficient set A_j by Equation 3;
  - Denote A^k as all coefficient sets $\{A_j\}$ $j = 1, \ldots, c$ and compute residual $r^k = \|Y - D^{k-1}A^k\|_F$;
  - **for** each $m \times m$ patch P_i^l on l_0 **do**
    - Reconstruct $y_i^r = D \cdot a_i$, where a_i is y_i's coefficients in A_j;
    - De-normalized $y_i^d = y_i^r \cdot v_i^c$;
    - Reconstruct normalized high-resolution feature vector $f_i^{h,r} = \text{de-concatenate}_{high}(y_i^d)$;
    - Reconstruct de-normalized feature vector $f_i^{h,d} = f_i^{h,r} \cdot v_i^h$;
    - Reconstruct high-resolution intensity patch $P_i^{h,r} = f_i^{h,d} + \text{mean}(P_i^h)$ where P_i^h is P_i^l's corresponding $mz \times mz$ patch on H_k;
  - Compute H^k = average of overlapped $P_i^{h,r}$;
  - Update dictionary D^k from D^{k-1} by Equation 4;
- Set $H = H^k$, where $k = \arg\min\{r^k\}$;

a corresponding high-resolution patch in I_k whose scaling factor is s. We use all the patch pairs to learn a dictionary with their group sparsity in order to capture the relationship among all the high-resolution or low-resolution patches, respectively.

In order to train this dictionary, we first extract features from low-resolution patches and high-resolution patches similar to [10]. The features we extract from low-resolution patch are two first-order image gradients and two second-order image gradients along horizontal and vertical axes, i.e. $[1, 0, -1]$, $[1, 0, -1]^\top$, $[-1, 0, 2, 0, -1]$, $[-1, 0, 2, 0, -1]^\top$. For each high-resolution patch, each feature vector is formed by raster scan of pixel values after subtracting the mean value of that patch.

For each high-resolution/low-resolution patch pair, we compose one concatenated feature vector. As the dimensions of low-resolution patch feature and high-resolution patch feature are different, we normalize both feature vectors independently in order to balance their contributions, before concatenating them into one single vector. All of the concatenated feature vectors are normalized to unit-norm vectors for dictionary learning with group sparsity constraints. Due to the feature design, it is possible that both of the high-resolution feature vector and low-resolution feature vector are zero. In such cases, these feature vectors are discarded.

To exploit the group similarity among patch pairs, we group pairs with similar feature vectors into clusters by K-means clustering. The feature we choose is the image gradient generated by low-resolution patches regardless of high-resolution patches because the low-resolution patches are more reliable than high-resolution patches.

With a given dictionary D, we solve the group sparse coefficients for each cluster U_i as

$$\min_{A_i} \|A_i\|_{1,2} \quad \text{s.t. } \|Y_i - DA_i\|_F \leq \sqrt{n_i}\delta, \tag{3}$$

where $\|A\|_{1,2} = \sum_{k=1}^{n} \|R^k\|_2$ and R^k is A's k-th row. In the equation above, Y_i is the column-wise feature vector in cluster U_i, n_i is the column number of Y_i, $\|\cdot\|_F$ is the Frobenius norm, and δ is a threshold controlling how similar the reconstructed feature vectors should be constructed from the original feature vectors. We use the SPGL1 package [16] to solve the above optimization problem.

As the group sparse coefficients are solved within separated cluster and the dictionary is given before solving the above equation, we need to update the dictionary for overall optimization. We denote A as the union of all coefficients A_i, and Y as the union of all feature vectors Y_i. The dictionary D is updated by the K-SVD algorithm [13],

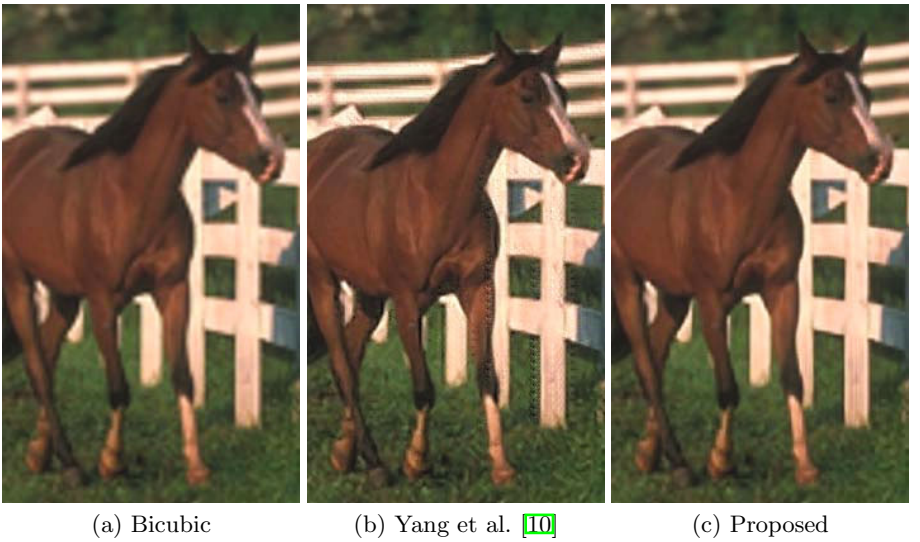
$$D = \arg\min_{D} \|Y - DA\|_F \quad \text{s.t. } \|D_j\|_2 = 1 \ \ \forall\, j, \tag{4}$$

where D_j is the j-th column of D. We iteratively solve group sparse coefficients in Equation 3 and Equation 4 until both A and D converge. The product of dictionary D and coefficient A contains the resulting feature vectors by patch similarity not only within each cluster but also among all clusters. We use these

feature vectors to generate the output high-resolution image. We summarize the process of this step in Algorithm 2.

3 Experimental Results

In this section, we describe the experimental setups and present the results using the proposed method and other algorithms. For all the experiments, we set the number of support low-resolution image $n = 6$, the number of nearest neighbor $m = 9$, variance of Gaussian blur kernel $\sigma^2 = 0.8$, scaling factor $s = 3$, and group sparse coding threshold $\delta = 0.05$. For a color input image, we convert it to YCbCr space and apply our algorithm only on luma component Y, and simply bicubic interpolate chroma components CbCr since human eyes are much more sensitive to luma rather than chroma. To compare with the state-of-the-art example-based algorithms, we use the original code provided by [10], and implement the algorithm of [12][1]. More results and MATLAB code can be found on `http://eng.ucmerced.edu/people/cyang35`



(a) Bicubic (b) Yang et al. [10] (c) Proposed

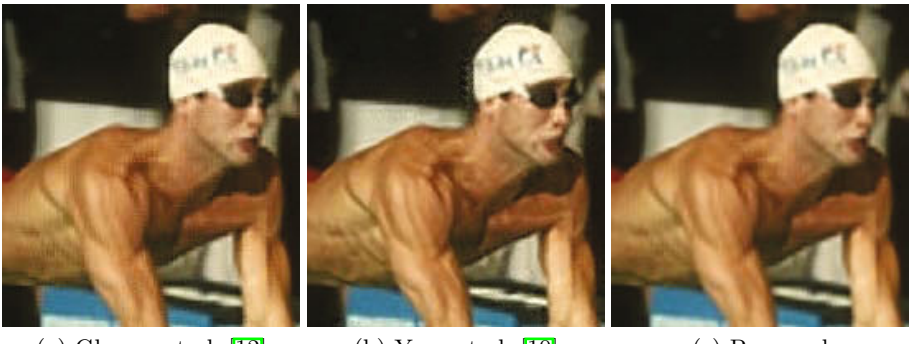
Fig. 2. Horse (results best viewed on a high-resolution display) Our result shows sharper edge than bicubic interpolation and less artifacts than [10] along fence and front legs.

We use images in the Berkeley segmentation dataset [17] for experiments. As shown in Fig. 2-7, the proposed algorithm generates shaper images with less

[1] This is based on our best efforts to implement the algorithm by Glasner et al. [12] with their help and suggestions as the authors do not release their code. The results may not be exactly the same as their reported results due to parameter settings.

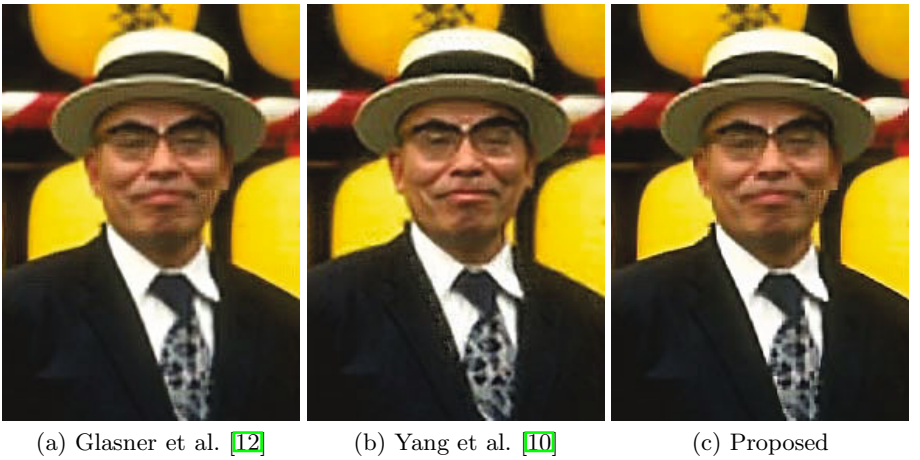
(a) Glasner et al. [12] (b) Yang et al. [10] (c) Proposed

Fig. 3. Deer (results best viewed on a high-resolution display). Compared with result generated [12], our super-resolution image has fewer artifacts (e.g., the antler region is smoother).Compared with result generated by [10], our super-resolution image has fewer artifacts (e.g., the antler region).



(a) Glasner et al. [12] (b) Yang et al. [10] (c) Proposed

Fig. 4. Swimmer (results best viewed on a high-resolution display). Compared with result generated by [12], our result has fewer artifacts (e.g., muscle and rib regions). Compared with result generated by [10], our result has fewer artifacts (e.g., around the head region).



(a) Glasner et al. [12] (b) Yang et al. [10] (c) Proposed

Fig. 5. Gentleman (results best viewed on a high-resolution display). Compared with result generated by [12], our result has less artifacts (e.g., on the forehead). Compared with result generated by [10], our result has less artifacts (e.g., on the collar region).

(a) Original (b) Proposed

(c) Yang et al. [10] (d) Glasner et al. [12]

Fig. 6. Boy (results best viewed on a high-resolution display). Compared with result generated by [12], our super-resolution image has fewer artifacts (e.g., several blotches in the facial and collar regions). Compared with result generated by [10], our super-resolution image has fewer artifacts (e.g., several large blotches in the lip and contour regions).

(a) Bicubic (b) Yang et al. [10] (c) Proposed

Fig. 7. Young Man (results best viewed on a high-resolution display). Our result shows sharper edge than bicubic interpolation and less artifacts than [10] along the collar and the stripes.

artifacts than the ones obtained by the example-based super-resolution algorithm [10]. Due to space limitation, we cannot present the full resolution images in this manuscript and these images are best viewed on high-resolution displays (additional results with high resolution images can be found in the supplementary material). For example, the super-resolution images generated by [10] have more artifacts along vertical strips or regions with intensity discontinuity, e.g., the horse legs in Fig. 2, the swimmer's cap in Fig. 4, the gentleman's collar in Fig 5, and the stripes in Fig. 7. In addition, the proposed algorithm outperforms the conventional super-resolution algorithm using bicubic interpolation. The results can be explained by the assumption of example-based super-resolution algorithm which entails the need to find matches between low-resolution and high-resolution image pairs from a large training set. However, this assumption does not always hold when the training set contains disparate images which are not directly relevant to the test image (i.e., the trade-off between generality and specialty). In contrast, our algorithm does not have this problem because the training set is constructed directly from the input frame rather than a fixed dictionary.

Compared with the results generated by [12], the super-resolution images by our method also have fewer artifacts, e.g., along antlers of the deer in Fig. 3 and facial regions around eyes and mouth in Fig. 6. The success of [12] depends on whether there are plentiful similar patches in the image pyramid generated by the input frame. For images with numerous repetitive patterns (e.g., sunflower fields or butterfly wings), this algorithm tends to work well. This algorithm is not expected to perform well for an image containing a unique object, e.g., a human standing in a natural scene as shown in Fig. 6. As this unique object occupies a relatively small region, this algorithm is not able to find a sufficient number of similar patches in the natural image using the low-resolution patches from the unique object (e.g., faces), and consequently produce improper high-resolution patches (i.e., generate super-resolution image patches of foreign objects). The resulting effects are especially noticeable as these unique objects are usually the focus of attention in these images. Our proposed algorithm does not have such artifacts because we exploit both of group similarity and patch similarity rather than mere patch similarity in [12]. Although the patches on human faces are few, they can be included in similar groups to maintain the similarity in the dictionary learning. Consequently, they produce much fewer artifacts in the super-resolution images.

4 Concluding Remarks

In this paper we propose an example-based super-resolution algorithm by exploiting self-similarities using one single input image. We exploit self-similarities on two fronts: both in generating image pairs and learning dictionary with group sparsity. Experimental results show our algorithm is able to achieve the state-of-the-art super-resolution images. Our future work will focus on algorithms that take the geometrical relationships among image patches into account for efficient and effective dictionary learning.

Acknowledgments. We would like to thank Daniel Glasner and Oded Shahar for numerous discussions regarding implementation details of their super-resolution algorithm.

References

1. Park, S.C., Park, M.K., Kang, M.G.: Super-resolution image reconstruction: A technical overview. IEEE Signal Processing Magazine, 21–36 (2003)
2. Morse, B., Schwartzwald, D.: Image magnification using level set reconstruction. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 333–341 (2001)
3. Fattal, R.: Image upsampling via imposed edge statistics. In: SIGGRAPH 2007: ACM SIGGRAPH 2007 papers. ACM, New York (2007)
4. Irani, M., Peleg, S.: Improving resolution by image registration. Computer Vision, Graphics and Image Processing 53, 231–239 (1991)
5. Lin, Z., Shum, H.Y.: Fundamental limits of reconstruction-based superresolution algorithms under local translation. IEEE Transactions on Pattern Analysis and Machine Intelligence 26, 83–97 (2004)
6. Freeman, W.T., Jones, T.R., Pasztor, E.C.: Example-based super-resolution. IEEE Computer Graphics and Applications, 56–65 (2002)
7. Sun, J., Zheng, N.N., Tao, H., Shum, H.Y.: Image hallucination with primal sketch priors. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 729–736 (2003)
8. Chang, H., Yeung, D.Y., Xiong, Y.: Super-resolution through neighbor embedding. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 275–282 (2004)
9. Sun, J., Sun, J., Xu, Z., Shum, H.Y.: Image super-resolution using gradient profile prior. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2008)
10. Yang, J., Wright, J., Huang, T., Ma, Y.: Image super-resolution via sparse representation of raw image patches. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2008)
11. Xiong, X., Sun, X., Wu, F.: Image hallucination with feature enhancement. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2009)
12. Glasner, D., Bagon, S., Irani, M.: Super-resolution from a single image. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 349–356 (2009)
13. Aharon, M., Elad, M., Bruckstein, A.: K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Transactions on Signal Processing 54, 4311–4322 (2006)
14. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Non-local sparse models for image restoration. In: Proceedings of IEEE International Conference on Computer Vision, pp. 2272–2279 (2009)
15. Arya, S., Mount, D.M.: Approximate nearest neighbor queries in fixed dimensions. In: SODA 1993: Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete algorithms, pp. 271–280 (1993)
16. Berg, E.v., Friedlander, M.P.: SPGL1: A solver for large-scale sparse reconstruction (2007), http://www.cs.ubc.ca/labs/scl/spgl1

17. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proceedings of IEEE International Conference on Computer Vision, pp. 416–423 (2001)

Appendix

Proof of Proposition 1: Assume $s_2 = s_1^n$, where n is a natural number and the subsample operator $\downarrow$ does not decrease image quality, then $I_{in} * B_2$ is equivalent to $((((I_{in} * B_1) \downarrow_{s_1}) * B_1) \downarrow_{s_1} \cdots * B_1) \downarrow_{s_1}$.

Also assuming the subsample operator can be ignored, it implies $I_{in} * B_2 = ((I_{in} * B_1) * B_1) \cdots * B_1$ n times. Using the associative law of a convolution operator in the discrete domain, i.e., $(f * g) * h = f * (g * h)$, it follows $I_{in} * B_2 = I_{in} * (B_1 * \cdots * B_1)$, and $B_2 = B_1 * \cdots * B_1$ n times.

Because we use Gaussian blur kernel and the convolution of two Gaussian kernels is still a Gaussian kernel whose variance is the sum of the two variances, i.e., $\sigma_2^2 = n \cdot \sigma_1^2$ as $B_2 = B_1 * \cdots * B_1$. With these equation together, $\sigma_2^2 = n \cdot \sigma_1^2$ and $s_2 = s_1^n$, it follows that $\sigma_1^2 = \sigma_2^2 \cdot \log(s_1)/\log(s_2)$. □

On Feature Combination and Multiple Kernel Learning for Object Tracking

Huchuan Lu[1], Wenling Zhang[1], 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. This paper presents a new method for object tracking based on multiple kernel learning (MKL). MKL is used to learn an optimal combination of χ^2 kernels and Gaussian kernels, each type of which captures a different feature. Our features include the color information and spatial pyramid histogram (SPH) based on global spatial correspondence of the geometric distribution of visual words. We propose a simple effective way for on-line updating MKL classifier, where useful tracking objects are automatically selected as support vectors. The algorithm handle target appearance variation, and makes better usage of history information, which leads to better discrimination of target and the surrounding background. The experiments on real world sequences demonstrate that our method can track objects accurately and robustly especially under partial occlusion and large appearance change.

1 Introduction

Visual tracking is an important task in many computer vision applications like visual surveillance, human computer interaction, and traffic monitoring or sports analysis. Tracking is significantly challenging because of the intrinsic appearance variability, and extrinsic illumination change. In order to ameliorate the adaptability of the algorithm to scene change, many different methods have been proposed for it. To adapt to the changes of object and background during tracking process, it is necessary to introduce on-line learning mechanism into trackers. Avidan's ensemble tracker [1] replaces some old weak classiers with new ones by AdaBoost at each frame. Grabner *etal* [2] adopt online AdaBoost firstly for feature selection and introduce selectors which paved the way for application on visual object tracking. In the paper [3,4], different online SVM algorithms are proposed and applied to object tracking. However, due to using results which the tracker gives to update the tracker itself, slight inaccuracies in the tracker can therefore lead to incorrectly labeled training examples, which degrades the classifier and may cause further drift. To resolve the problem, Avidan [1]adopts a simple outliers rejection scheme, and Babenko *etal.* [5]introduce the concept of Multiple Instance Learning into visual tracking. Also these methods are inclined to fail in case of partial or complete occlusion, for they rely on the global template model focusing on the integrated appearance information of the object.

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 511–522, 2011.

Yin *etal.* [6] consider visual object tracking as a numerical optimization problem by making a numerical hybrid local and global mode-seeking tracker that combines detection and tracking. "Frag-Track" [7] extracts the integral histogram of multiple image fragments or patches to represent the template, calculates each patch voted by comparing its histogram with the corresponding image patch histogram and then combines voted maps of the multiple patches. Experiments demonstrate that the approach can deal with partial and swift occlusion well. In addition, some works are also tried to resolve the occlusion problem in the multiple objects tracking.

Recently multiple kernel learning (MKL) methods [8,9]have shown great advantages in various classification(e.g. Learning the discriminative power-invariance trade-off; Support kernel machines for object recognition). Instead of using a single kernel in support vector machine (SVM) [10], MKL learns an optimal kernel combination and the associated classifier simultaneously, and provides an effective way of fusing informative features and kernels. However, these methods basically adopt a uniform similarity measure over the whole input space. When a category exhibits high variation as well as correlation with other categories in appearance, they are difficult to cope with the complexity of data distribution. Varma *etal.* [11] proposed combining multiple descriptors using Multiple Kernel Learning(MKL) and showed impressive results on varied object classification tasks. So in this paper, we propose Multiple Kernel Learning for Tracking (MKLT) approach uses an easily obtained training data as input, and then tunes itself to the classification for tracking at hand. It simultaneously updates the training examples to tailor them towards the objects in the scene. It also updates the weights that determine the optimal combination of different kernels, while allowing different combinations to be chosen for different objects. Finally, it tunes the classifier to the updated training data. Our final system is obtained by combining the outputs of this online classifier with the high probability outputs of the original classifier trained on the first frames.

We firstly describe the MKL formulation of Bach*etal.* [12]. More efficiency in multiple kernel learning known as SimpleMKL [13], which we use to obtain a classifier for initial training frames. SimpleMKL carries out this optimization in an SVM framework to learn the SVM model parameters as well as kernel combination weights simultaneously. Our tracking procedure with MKL is an exacting online solution that allows us to update the Lagrangian multipliers of the training data, as well as the kernel combination weight, five or ten frames at a time. The main contribution is that we adopt spatial pyramid [14] to obtain the features' spatial information which is inspired by the remarkable ability of "bag of words" to handle intra-class pose variant and occlusion. The spatial pyramid works by computing rough geometric correspondence on a global scale using an efficient approximation technique adapted from the pyramid matching scheme of Grauman and Darrel [15]. However, the satisfactory results are obtained by our approach when occlusions occur or the object' length and scale change severely, as well as the ability to recapturing the object.

2 Related Work

Early works on an object feature extracting used global features such as color or texture histogram [16]. However, these features were not robust to view-point changes, clutter and occlusion. Over the years, more sophisticated approaches such as part-based [17]and bag of features [18]methods have become more popular. Increased interest in object recognition has resulted in new feature descriptors and a multitude of classifier. Inspired by the pyramidal feature matching approach of [15], Bosch *et al.* proposed two new region descriptors - the Pyramid Histogram of Oriented Gradients (PHOG) and Pyramidal Histogram of Visual Words (PHOW) [19]zhang *etal.* used the Geometic Blur(GB) feature [20]and proposed using a discriminative nearest neighbor classification for object recognition. Wu *etal.* [21]used edge features to capture the local shape of objects.

Kernel based method is one of attractive research areas for object categorization in recent years. Diverse kernels such as pyramid matching kernel(PMK) [15], spatial pyramid matching kernel(SPK) [14], distribution kernel (PDK) [22] and chi-square kernel are delicately designed to compute the similarity of image pair on certain features that represent particular visual characteristics. Multi-kernel based classifiers have been introduced into object categorization yielding promising results. And multiple features (e.g. appearance, shape) are employed and kernels (e.g. PMK and SPK with different hyper-parameters) are linearly combined in MKL framework.

Lanckriet *etal* [23]introduced the MKL procedure to learn a set of linear combination weights, while using multiple features of information with a kernel method, such as an SVM. It can result in a convex but non-smooth minimization problem. The algorithm worked for hundreds of examples or hundreds of kernels [12] provided the additional advantage of encouraging sparse kernel combinations. Our initial object classifier built the object's features from the first few frames. Our work builds on MKL and fits well into the SVM framework. And also we provide the online updating process to retrain the classifier that accounts for appearance changes and allows reacquisition of an object after total occlusion.

3 Algorithms

3.1 The Multiple Kernel Learning

Kernel based learning methods have proven to be an extremely effective discriminative approach to classification as well as regression problem. One approach performing kernel selection is to learn a kernel combination during the training phase of the algorithm. One prominent instance of this class is MKL. Its objective is to optimize jointly over a linear combination of kernel(equation1). Given multiple features, one can calculate multiple basis kernels, one for each feature. So the kernel is often computed as a convex combination of the basis kernels,where x_i are objects' some samples, $k_m(x_i, x_j)$ is the m^{th} Kernel, and d_m are the weights given to each kernel.

$$K(x_i, x_j) = \sum_{m=1}^{M} d_m \, k_m(x_i \, , x_j) \; , \; \sum_{m=1}^{M} d_m = 1 \; , d_m \geqslant 0 \tag{1}$$

Learning the classifier model parameters and the kernel combination weights in a single optimization problem is known as the Multiple Kernel Learning problem [23]. There are a number of formulations for the MKL problem, our approach builds on the MKL formulation of [13], known as SimpleMKL. This formulation enables the kernel combination weights to be learnt within the SVM framework. The optimization equation is given by equation 2,3,4:

$$\min \quad \sum_{m} \frac{1}{d_m} \, w_m \, w_m^T + C \sum_{i} \xi_i \tag{2}$$

$$such\, that \;\; y_i \sum_{m} \varphi_m(x_i) + y_i \, b \geqslant 1 - \xi_i \;\; \forall i \tag{3}$$

$$\xi_i \geqslant 0 \;\forall i, \;\; d_m \geqslant 0 \;\forall m, \;\; \sum_{m} d_m = 1 \tag{4}$$

Where b is the bias, ξ_i is the slack afforded to each sample data and C is the regularization parameter. The solution to the MKL formulation is based on a gradient descent on the SVM objective value.

The final binary decision function of MKL is of the following form:

$$F_{MKL}(x) = sign(\sum_{m=1}^{M} \beta_m (k_m \, (x)^T \, \alpha + b)) \tag{5}$$

The only free parameter in the MKL approaches is the regularization constant C, which is chosen using Cross Validation (CV). In this paper we study a class of kernel classifiers that aim to combine several kernels into a single model. We associate object features (color features, spatial pyramid histogram features) with different parameters of kernels (Gauss kernel, χ^2 kernel) functions, kernel combination/selection translates naturally into feature combination/selection.

$$k(x_1, x_2) = \chi^2(x_1, x_2) \quad \chi^2(x_1, x_2) = \sum \frac{(x_1 - x_2)^2}{(x_1 + x_2)} \tag{6}$$

$$k(x_1, x_2) = \exp((-(x_1 - x_2)^2)/(2\sigma^2)) \tag{7}$$

A conceptually simple approach is the use of CV to select the different parameters of kernels.

3.2 Spatial Pyramid Histogram Feature

The traditional bag of features methods, which represent an image as an orderless collection of local features, have severely limited descriptive ability, because there methods disregard all information about the spatial layout of features. Lazebink

etal. [14] provides repeatedly subdividing the image and computing histograms of local features at increasingly fine resolutions. The spatial pyramid framework suggests a possible way to address the issue: namely, the best results may be achieved when multiple resolutions are combined in a principled way.

We describe the original formulation of pyramid matching [15]. Consider matching two images each consisting of a 2D point set, where we wish to determine matches between the point sets when the images are over laid for a particular point the strength of the match depends on the distances from its position to points in the other set. Each image is divided into a sequence of increasingly finer spatial grids by repeatedly doubling the number of divisions in each axis direction. The number of points in each grid cell is then recorded. This is a pyramid representation because the number of points in a cell at one level is simply the sum which is divided into four cells at the next level. The cell counts at each level of resolution are the bin counts for the histogram representing that level. The correspondence between the two point sets can then be computed as a weighted sum over the histogram intersections at each level. Similarly, the lack of correspondence between the point sets can be measured as a weighted sum over histogram differences at each level. It is illustrated in Fig.1.

A spatial pyramid histogram is the single histogram intersection of "long" vectors which is formed by concatenating the appropriately weighted histograms of all cells at all resolutions. In the long histogram, the three color bins denote features' bins of image and the height expresses occurrence which extracted features fall in each bin. When L=0, the histogram is the first part (level 0 - there are only three bins). We can see details in [14].

3.3 Online Updating

There is a set of data samples $(x_1, x_2, \ldots, x_n)$ with corresponding class labels $(y_1, y_2, \ldots, y_n)$. Let $\Phi_k(x_i, x_j)$ be the set of K kernels. The MKL solution for the

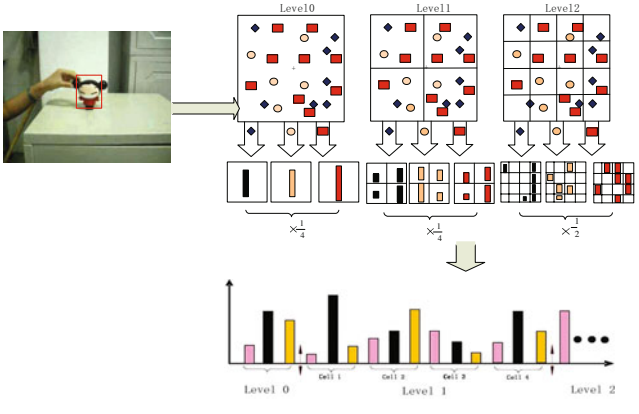


Fig. 1. Example of constructing a three-level spatial pyramid histogram. We weight each spatial histogram and concatenate them to form a long histogram.

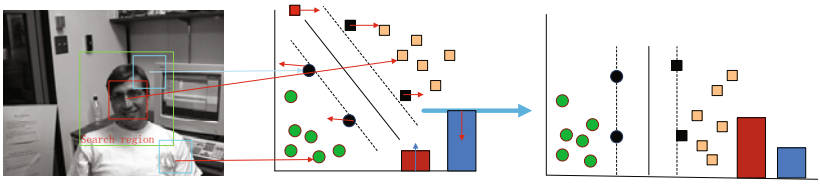


Fig. 2. (a):The object region,search region and the context region.(b):The black circles and rectangles are support vectors.kernel 1 (brown bar) kernel 2(red bar).(b)shows the effect of adding a new sample (shown in red) on the original samples and weights. Some samples change.(c)shows the final classifier after adding a new sample; and the corresponding weights are changed.

given data is obtained by SimpleMKL [13]. The data samples are divided into three disjoint sets based on their Lagrange multipliers: lying on the correct side of the margin vectors ($\alpha_i = 0$) support vectors ($0 < \alpha_i < C$) and lying on the wrong side of the margins ($\alpha_i = C$).In this paper, we adopt the online updating the tracker to make it adapt to the appearance change of the target. Through the above, a classifier based MKL is formed. When a new object sample is added to the solution, we need to calculate its Lagrange multiplier ($0 \leqslant \alpha_t \leqslant C$) such that the KKT conditions are satisfied once again by using the support vectors last time and the new samples. In the process of solving the problem, the kernel weights and the bias will be changed to maintain the constraints in KKT. It is shown in Fig.2 in short. A new point marked in red is added to the system. In order to adopt the KKT conditions, the margin changes, while some of the other points change set membership. At the same time, the kernel combination weights also change. The KKT conditions for our problem are derived from the Lagrangian function corresponding to equation 8.

$$\begin{aligned} L &= \frac{1}{2}\sum_m \frac{w_m\, w_m}{d_m} + C\sum_i \zeta_i - \sum_i v_i\,\zeta_i - \mu_m\, d_m - A - \lambda(\sum_m d_m - 1) \\ A &= \sum_i \alpha_i(y_i\, w_m\, \phi_m(x_i) + y_i\, b - 1 + \zeta_i) \end{aligned} \tag{8}$$

4 Our Tracking Framework

Within the context of object tracking, we define the object region and its surroundings as positive samples and negative samples respectively, as shown in Fig.2(a). Our target is to learn a MKL classifier which can classify the positive sample and negative sample in the new frame. Starting from first few frames, the positive and the negative samples are used to training the MKL classier. Then the search region can be estimated in the next frame. Finally, the target region

in next frame is located with local maximum score within the search region.The incremental tracking [24] performs as guidance in our whole tracking process.

In order to improve the discriminative power, we utilize higher dimensional "strong features"–spatial pyramid histogram and color histogram. We use a dense regular grid instead of interest points' detection to extract SIFT features because the former can capture uniform regions such as forest, face. And the SIFT descriptors are then vectors quantized into visual "words" for the dictionary. The vector quantization is carried out by K-means clustering algorithm. Therefore, an image can be represented as a histogram which is equivalent to the occurrence frequency of dictionary of a sample. SIFT features are extracted by a dense regular grid technique, and are multi-image representations of an image neighborhood. They are Gauassian derivatives computed at 8 orientation planes over a 4×4 grid of spatial locations, giving a 128-dimension vector. The color histogram is well known so we don't need to introduce it.

One of the most difficult tasks for a tracker is how to online update the tracker to make it adapt to the appearance change of the target. Here we propose a

Algorithm 1. Online MKL Tracking & Updating

Input: I_n Video frames for processing
Output: Rectangles of target object's region
Training with the first frames $I_n(n = 10+)$**:**
(1) Manually initializing parameters describing the property of region of interest (center, size and rotation angle) in the first frame.
(2) Parameters sampling and considering the optimization produced by IVT[19] model as positive samples in the first few frames. Also obtaining negative samples around the positive samples.
(3) Extracting features (color histogram and spatial pyramid histogram) from the positive and negative samples.
(4) Train the MKL classifier (in the processing of kernel calculation, the Gaussian kernels are computed on the color features and χ_2 kernels are computed on the spatial pyramid histogram features) to get $F_{MKL}(x) = sign(\sum_{m=1}^{M} \beta_m (k_m(x)^T \alpha + b))$, and its support vectors $V_1 = \{x_i, y_i\}_{i=1}^{M}$
Online tracking: When a new frame comes:
(1) Randomly sampling 300 candidates with different affine parameters around the object's position obtained in the previous frame.
(2) Extract features, using the trained MKL classifier to find the maximum score (the object's location) given by $F_{MKL}(x)$ and go to the next frame.
(3) Approximately accumulate five or ten frames, update the MKL classifier (refresh positive samples $p_n = V^+ \cup C^+$ and negative samples $N_n = V^- \cup C^-$ V is the support vectors, and C are the new positive and negative samples by tracking).
(4) Retrain the MKL classifier using new samples for updating to $F_{MKL}(x) = sign(\sum_{m=1}^{M} \beta_m (k_m(x)^T \alpha + b))$
Go to next frame.
End

simple yet effective way for on-line updating the linear MKL classifier which is inspired by the [25]. And our tracker can not only record the "Key Frames"(first few frames) of the target as the history information, but can also update online to decrease the risk of drift. By online updating, the MKL tracker can adjust its hyper-plane for the maximum margin between the new positive and negative samples. The support vectors transferred frame by frame contain important "Key Frames" of the target object in the previous tracking process. The details are described as Algorithm 1.

We adopt the random sampling for improving efficiency rather than sliding window technique: in the following frame, a number of candidates with different affine parameters are generated according to a Gaussian distribution centered at the central position of the target in the previous frame. Incremental PCA tracking [24] is applied in order to collect sufficient training samples in the first few frames. In each frame, we obtain one optimal tracking result as the positive sample according to incremental PCA [24]. At the same time, a few negative samples are randomly selected around the optimal target. Note that the sampling radius should be properly predefined to guarantee that negative samples cannot overlap with the positive ones. While combining features is very beneficial, the gain obtained by MKL over simple one kernel or averaging is modest. However MKL determines a sparse selection of features, which helps improve the efficiency of inference.

5 Experimental Results

Our tracking framework is implemented on a Pentium Dual Core 1G PC with 1G memory. All the test sequences are 320× 240 resolutions. A 128-dim SIFT feature is extracted to represent objects at first. The size of dictionary which we have formed is 20. And the spatial pyramid's level is 3. Thus an object is represented as a 420-dim vector (spatial pyramid histogram) by the method proposed. The

Fig. 3. "sail"sequence Top row: results of RGBSVM. Middle row: results of pyramidSVM. Bottom row: results of our approach.

Fig. 4. "singer1"sequence Top row: results of RGBSVM. Middle row: results of pyramidSVM. Bottom row: results of our approach.

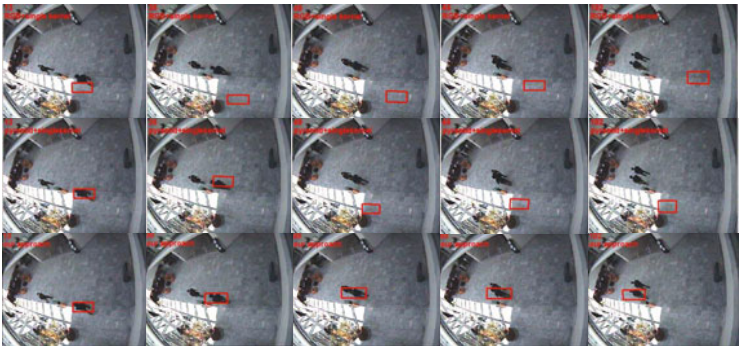


Fig. 5. "Meetwalksplit"sequence Top row: results of RGBSVM. Middle row: results of pyramidSVM. Bottom row: results of our approach.

object's color histogram is a 256-dim vector. We sample 300 candidates randomly around the tracked object in the previous frame in the tracking. All participating kernels are Gaussian kernels and χ^2 kernels. There are 50 different parameters of Gaussian kernels (in the range from 5 to 250) and χ^2 kernels respectively (in the range from 10 to 500). For MKL we fix C = 1000 which yields best results. In our experiments, we compare the results of the color information with the single kernel (RGBSVM), spatial pyramid information with the single kernel (pyramidSVM) and our approach (color+spatial information with MKL)on publicly available datasets and our own datasets. In Fig.3(sequence "sail"–our datasets), the boy's face is occluded by a book constantly. At the beginning all the three approaches can keep track of the face, but RGBSVM gradually fails when encountering complex and long time occlusion. At the same time, pyramidSVM can track the object reluctantly. But it is not stable and it drifts away the face. On the contrary, our approach successfully tracks the face during the entire tracking process. Fig.4 shows the tracking results for the "Singer1"sequence

Fig. 6. "EnterExitCrossingPaths1cor"sequence Top row: results of RGBSVM. Middle row: results of pyramidSVM. Bottom row: results of our approach.

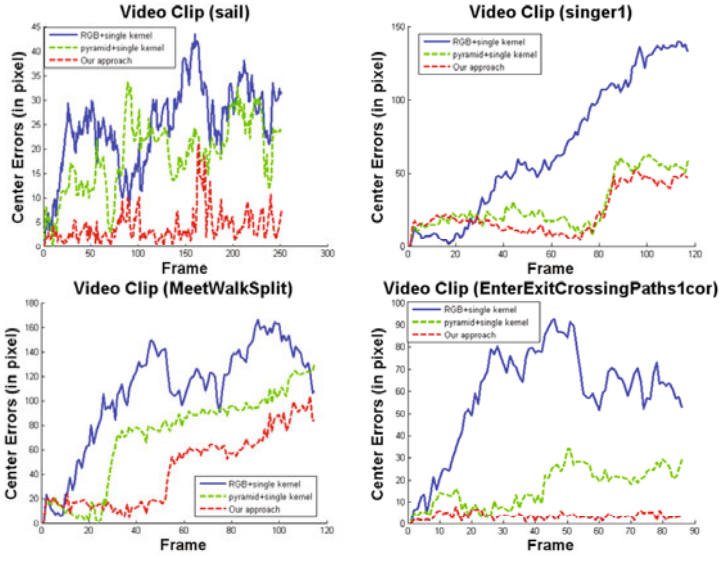


Fig. 7. The four videos' quantitive comparisons among the results of the three approaches –our approach(red), RGBSVM(blue) and pyramidSVM(green)

which is from Junseok Kwon *etal.* [26]. This sequence includes scale change and large illumination changes. The RGBSVM can not lock the object completely when there are illumination changes but pyramidSVM track the object all the same. We deduce that the spatial pyramid histogram feature displays better than the generally features. Our approach combines color, global and local feature (spatial pyramid histogram), adds updating mechanism, keeps the target still and is more robust. Fig.5, in the "Meetwalksplit"video which is from the

CAVIAR database1, both RGBSVM and pyramidSVM display badly, looses the target completely after frame 30. The video exhibits challenges, including occlusions, the small target and fast motion (which causes motion blur). But our approach performs the best. Another result -"EnterExitCrossingPaths1cor"(It is also from the CAVIAR database1) Fig.6 also shows the compared results. Though the experiment results, the approach which uses multiple kernels can be performed better than the single kernel approach, because we combine different features and different kernels.Different from existing approaches that use a linear weighting scheme to combine different features, our approach does not require the weights to remain the same across different samples, and therefore can effectively handle features of different types with different kernels. Fig.7 is the previous four videos' quantitive comparisons among the results of the three approaches–our approach, RGBSVM and pyramidSVM. RGBSVM and pyramidSVM perform not well mostly. In contrast, our approach is robust in handling occlusion,scaling and illumination changing.

6 Conclusions

In this paper, we propose a tracking framework successfully incorporating "spatial pyramid histogram" and Multiple Kernel Learning(MKL) to deal with occlusions and illumination changes, scale changes. We adopt IVT algorithm to collect training samples to training the MKL classifier using color feature and spatial pyramid histogram in the first few frames. The most important part in our paper is that we apply the spatial pyramid method to partition the image into increasingly fine sub-regions to construct long weighted and jointed histogram which includes the images' spatial information. The spatial pyramid can represent the object globally and locally. An updating mechanism –online Multiple Kernel Learning classier is adopted to deal with pose and appearance changes of object. Experiments show that our approach outperforms RGBSVM in handling occlusions and pyramidSVM in handling scaling and rotation. In a word, our approach is more robust in various situations than other methods.

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. Avidan, S.: Ensemble tracking. In: CVPR (2005)
2. Grabner, H., Bischof, H.: On-line boosting and vision. In: CVPR (2006)
3. 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)
4. Tang, F.F., Brennan, S.: Co-tracking using semi-supervised support vector machines. In: ICCV (2007)

5. Babenko, B., Yang, M.-H.: Visual tracking with online multiple instance learning. In: CVPR (2009)
6. Yin, Z., Robert, T.: Collins object tracking and detection after occlusion via numerical hybrid local and global mode-seeking. In: CVPR (2008)
7. Adam, A., Rivlin, E., Shimshoni, I.: Robust fragmentsbased tracking using the integral histogram. In: CVPR (2006)
8. Bach, F.R., Lanckriet, G.R.G., Jordan, M.I.: Multiple kernel learning, conic duality, and the smo algorithm. In: ICML (2004)
9. Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: CIVR (2007)
10. Platt, J.: Fast training of support vector machines using sequential minimal optimization. In: Advances in Kernel Methods - Support Vector Learning. MIT Press, Cambridge (1998)
11. Varma, M., Ray, D.: Learning the discriminative power invariance trade off. In: ICCV (2007)
12. Rakotomamonjy, A., Bach, F., Canu, S.: More efficency in multiple kernel learning. In: ICML (2007)
13. Rakotomamonjy, A., Bach, F.R., Canu, S., Grandvalet, Y.: Simplemkl. Journal of Machine Learning Research 9, 2491–2521 (2008)
14. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR (2006)
15. Grauman, K., Darrell, T.: Pyramid match kernels: Discriminative classification with sets of image features. In: ICCV (2005)
16. Pontil, M., Verri, A.: Support vector machines for 3d object recognition (PAMI)
17. Fergus, R., Perma, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: CVPR (2003)
18. Sivic, J., Russell, B., Efros, A., Zisserman, A., Freeman, W.: Discovering objects and location in images. In: ICCV (2005)
19. Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: CIVR (2007)
20. Bosch, A., Zisserman, A., Munoz, X.: Geometric blur for template matching. In: CVPR (2001)
21. Wu, B., Nevatia, R.: Simultaneous object detection and segmentation by boosting local shape feature based classifier. In: CVPR (2007)
22. Haibin, L., Soatto, S.: Proximity distribution kernels for geometric context in category recognition.In: ICCV (2007)
23. Lanckriet, G., Cristianini, N., El Ghousi, L., Bartlett, P., Jordan, M.: Learning the kernel matrix with semi-definite programming. JMLR (2004)
24. Lim, J., Ross, D., Lin, R.-S., Yang, M.: Incremental learning for visual tracking. In: NIPS (2004)
25. Matthews, I., Ishikawa, T., Baker, S.: The template update problem. PAMI 26, 810–815 (2004)
26. Kwon, J., Lee, K.M.: Visual tracking decomposition. In: CVPR (2010)

Correspondence-Free Multi Camera Calibration by Observing a Simple Reference Plane

Satoshi Kawabata and Yoshihiro Kawai

National Institute of Advanced Industrial Science and Technology (AIST)
AIST Tsukuba Central 2, 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan

Abstract. In the present paper, we propose a multi camera calibration method that estimates both the intrinsic and extrinsic parameters of each camera. Assuming a reference plane has an infinitely repeated pattern, finding corresponding points between cameras is regarded as being equivalent to the estimation of discrete 2-D transformation on the observed reference plane. This means that the proposed method does not require any overlap of the observed region, and bundle adjustment can be performed in the sense of point-to-point correspondence. Our experiment demonstrates that the proposed method is practically admissible and sufficiently useful for building a simple shape measurement system using multiple cameras.

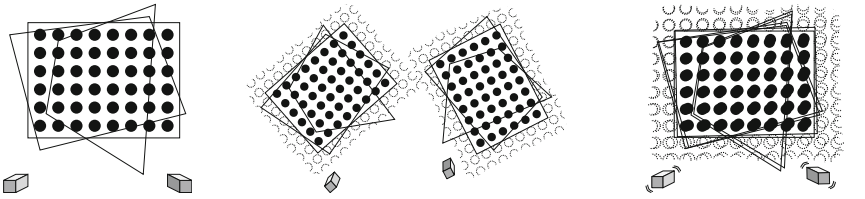
1 Introduction

Camera calibration is a fundamental problem in computer vision. As such, a great deal of research has been conducted on this topic. In the present paper, we propose a practical multi camera calibration method using a simple reference plane without any previous knowledge about the global correspondence between camera images and points on the reference plane. The proposed method is similar to Zhang's calibration for a single camera [1] except that the proposed method also estimates the relative positions and orientations (referred to hereinafter as 'pose') of the synchronized cameras. The calibration process is quite simple: users simply take several images synchronously using multiple cameras while moving the reference plane in front of the cameras.

Generally, relative pose estimation requires corresponding points between the camera images. In order to obtain these points, users often use a complex pattern that is easily detectable with a computer or simply impose some restrictions on the alignment of cameras such that, for example, the directions of all cameras are similar or the fields of view of the cameras generally overlap.

In contrast, the proposed method does not require such correspondence beforehand. Assuming an (infinitely) repeating pattern in a reference plane, correspondence detection can be regarded as the estimation of a discrete 2-D transformation within assigned temporal coordinates of the reference plane observed by each camera, as described later herein. This enables not only rough relative pose estimation based on geometry, but also enables bundle adjustment

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 523–534, 2011.



(a) Observation planes (b) Individual camera calibration (c) Merger and optimization

Fig. 1. Proposed calibration steps

to be performed in the sense of point-to-point correspondence. The proposed calibration method consists of the following steps (Fig. 1):

1. Single camera calibration (Zhang's method) of each camera,
2. Estimation of the relative positions of each camera using geometric constraints,
3. Estimation of correspondence between camera images,
4. Bundle adjustment of the corresponding points.

There are three categories of calibration methods when grouped according to the type of reference object: (a) a 3-D reference object, (b) a reference plane, and (c) natural features (self-calibration). Using a 3-D reference object allows us to directly compute a camera parameter from a set of sufficient pairs of 3-D reference points on the object and its image point [2]. This technique requires precisely created 3-D objects. In contrast, self-calibration does not use reference objects [3]. Although self-calibration is a sophisticated technique, this technique cannot determine the scale of a scene. Furthermore, it requires corresponding points, which generally reside in a region that can be observed by another camera. This leads to over-fitting of the region.

The plane-based method uses a reference plane instead of a 3-D object but is flexible in that the plane can be moved arbitrarily by hand. In addition, scale information can be recovered using knowledge of the pattern on the reference plane. Thus, the plane-based method is suitable for practical applications that require scale information. However, since the method estimates only the intrinsic parameter of a single camera, relative poses must be determined by another method when using a multi camera system [4]. Geometrically, such relative poses can be estimated from the set of poses of the reference planes [5]. However, bundle adjustment is not performed because of the lack of information on point-to-point correspondence.

Ueshiba et al. proposed a multi camera calibration method using a lattice patterned reference plane based on the factorization method [6]. Although this method can recover relative poses using corresponding points between camera images, the method by which to obtain such a correspondence was not described. Ramalingam et al. proposed a plane-based calibration for a generic camera (including multi camera systems) [7]. Although this method does not require correspondence, it does requires overlapping of the reference planes of each camera

view. Another solution to the problem of correspondence detection is to encode coordinate information into the pattern [8]. However, this approach has problems. First, creating a pattern without special software is difficult. Moreover, pattern recognition is more complicated. Finally, dense arrangement of patterns is not possible due to the requirement of a certain marker size.

2 Single Camera Calibration Using a Reference Plane

2.1 Zhang's Method [1]

The relationship between a 3-D point and its image point in a pinhole camera model is represented as

$$\lambda\tilde{\boldsymbol{x}} = A[R\,|\,\boldsymbol{t}]\tilde{\boldsymbol{X}} \tag{1}$$

where $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{X}}$ are the homogeneous vectors of $\boldsymbol{x} \in \mathcal{R}^2$ and $\boldsymbol{X} \in \mathcal{R}^3$, respectively, A is a 3×3 upper triangular matrix of the intrinsic parameter, and $[R\,|\,\boldsymbol{t}]$ is a 3×4 matrix of the extrinsic parameter consisting of a rotation matrix R and a translation vector $\boldsymbol{t}$. Practically, an image captured by a real camera is distorted nonlinearly by the lens. In order to eliminate this effect, the imaging process may be extended by adding two (or more) parameters, as follows:

$$\begin{cases} \breve{x} = x + x\left(\kappa_1\left(x^2+y^2\right) + \kappa_2\left(x^2+y^2\right)^2\right) \\ \breve{y} = y + y\left(\kappa_1\left(x^2+y^2\right) + \kappa_2\left(x^2+y^2\right)^2\right) \end{cases} \tag{2}$$

where (x, y) and $(\breve{x}, \breve{y})$ are the ideal and distorted coordinates, respectively, in the normalized coordinate system, and κ_1 and κ_2 are coefficients for modeling the (radial) distortion [1].

Homography H between a reference plane and an image plane is described as follows:

$$\lambda\tilde{\boldsymbol{x}} = A[\boldsymbol{r}_1\ \boldsymbol{r}_2\ \boldsymbol{t}][X\ Y\ 1]^{\mathrm{T}} = H[X\ Y\ 1]^{\mathrm{T}}. \tag{3}$$

In the closed form of Zhang's calibration, the relationship between the intrinsic parameter and the homography matrix is given as follows:

$$\begin{bmatrix} \boldsymbol{v}_{12}^{\mathrm{T}} \\ \boldsymbol{v}_{11} - \boldsymbol{v}_{22} \end{bmatrix} \boldsymbol{b} = \boldsymbol{0} \tag{4}$$

where

$$B = \{B_{ij}\} = A^{-\mathrm{T}}A^{\mathrm{T}} \tag{5}$$

$$\boldsymbol{b} = [B_{11},\, B_{12},\, B_{22},\, B_{13},\, B_{23},\, B_{33}]^{\mathrm{T}} \tag{6}$$

$$\begin{aligned} \boldsymbol{v}_{ij} = [&h_{i1}h_{j1},\, h_{i1}h_{j2} + h_{i2}h_{j1},\, h_{i2}h_{j2}, \\ &h_{i3}h_{j1} + h_{i1}h_{j3},\, h_{i3}h_{j2} + h_{i2}h_{j3},\, h_{i3}h_{j3}]^{\mathrm{T}}. \end{aligned} \tag{7}$$

By arranging the above equations for each observation into V, we obtain the following linear equation:

$$V\boldsymbol{b} = \boldsymbol{0}. \tag{8}$$

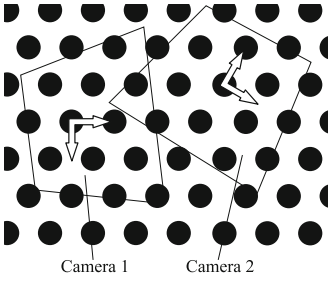


Fig. 2. Assigned coordinates of each camera

Next, the intrinsic parameter can be computed from this solution, and the extrinsic parameter is computed by $\lambda' A^{-1} H_i = [\boldsymbol{r}_{i1}\ \boldsymbol{r}_{i2}\ \boldsymbol{t}_i]$. This extrinsic parameter describes the relative pose between the camera and the observed reference plane.

The initial value of the distortion parameters can then be estimated, if desired, followed by nonlinear optimization, which minimizes the reprojection error:

$$\sum_{i,j} ||\boldsymbol{x}_{ij} - \breve{\boldsymbol{x}}(A, \kappa_1, \kappa_2, R_i, \boldsymbol{t}_i, \boldsymbol{X}_j)||^2 \tag{9}$$

where $\breve{\boldsymbol{x}}(\cdot)$ is the projection of a 3-D point $\boldsymbol{X}_j$ at frame i to the image plane. In our implementation, we adopt the weighed least squares method rather than Eq. (9) in order to handle the reliability of point detection.

2.2 Corresponding Points between Two Cameras

In the homography-based calibration of a single camera, the origin or direction of axes assigned to a reference plane at each observation need not be considered, because only the pose and scale of the reference plane are meaningful. However, the correspondence of the coordinates is required for computing the relative pose of two (or more) cameras. If an observed scene is sufficiently complex, such correspondence is obtained by comparing a feature of points [9], although this is difficult for a simple tiled pattern, which tends to have several similar features.

Next, consider taking images of a reference plane using two cameras and arbitrarily assign coordinates to each of the observed planes. Then, the relationship between these coordinates is represented by a transformation on a 2-D plane (Fig. 2).

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} X' \\ Y' \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{10}$$

Using the knowledge that the reference pattern contains a repeated pattern, in the case of Fig. 2, the transformation can be restricted to

$$\begin{bmatrix} t_x \\ t_y \end{bmatrix} = a \begin{bmatrix} 1 \\ 0 \end{bmatrix} + b \begin{bmatrix} \frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{bmatrix}, \quad \theta = c \cdot \frac{\pi}{3}\ [\mathrm{rad}] \tag{11}$$

where a, b, and c are arbitrary integers, and we assume the interval length to be 1. Therefore, when we obtain an approximate transformation, we can exactly

match the coordinates of each camera by simply rounding the approximate values to integers. Then, all of the points on a reference plane observed by one camera correspond exactly to points observed by the other camera.

As mentioned above, matching points on a repeated pattern is equivalent to estimating a certain discrete 2-D transformation. Thus, conditions such as the condition by which corresponding points must be observed from all cameras are no longer required. In the next section, we describe the method used to estimate this 2-D transformation using the relative poses of the reference plane at each observation of each camera obtained through Zhang's calibration.

3 Multi Camera Calibration

After applying Zhang's calibration to n cameras observing m planes, we obtain n sets of intrinsic parameters and nm relative poses of reference planes:

$$\{A_i\}_{i=1}^{n}, \quad \{R_{ij}, \boldsymbol{t}_{ij}\}_{j=1..m}^{i=1..n}. \tag{12}$$

Note that each relative position vector $\boldsymbol{t}_{ij}$ generally does NOT represent the same point on the reference plane for any combination of (i, j). Because Zhang's calibration requires only a homography set, the user can freely assign an arbitrary 2-D coordinate system to each plane. We refer to this coordinate system as the *temporal coordinate system* (Fig. 3).

3.1 Observation of a Reference Plane

As we described in Subsection 2.2, temporal coordinates can be represented by a combination of translation and rotation on a 2-D plane of a certain base coordinate system (i.e., *global coordinate system*). Therefore, the extrinsic parameters $\{R_{ij}, \boldsymbol{t}_{ij}\}_{j=1..m}^{i=1..n}$ obtained through Zhang's calibration contain the following:

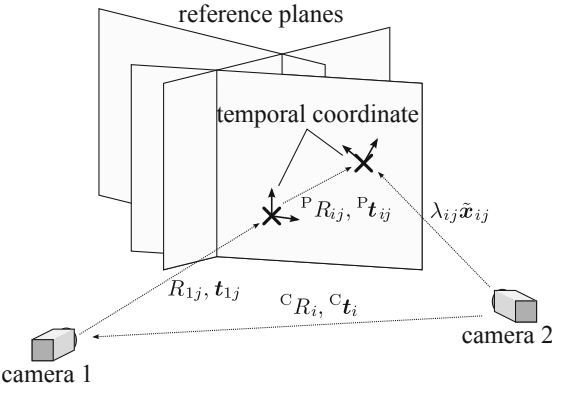


Fig. 3. Coordinate systems

– relative pose of the i-th camera ($[{}^{\mathrm{C}}R_i \ {}^{\mathrm{C}}\boldsymbol{t}_i]$),
– relative pose of a reference plane at j ($[{}^{\mathrm{R}}R_j \ {}^{\mathrm{R}}\boldsymbol{t}_j]$),
– discrete 2-D transformation of the temporal coordinates ($[{}^{\mathrm{P}}R_{ij} \ {}^{\mathrm{P}}\boldsymbol{t}_{ij}]$).

Thus, the entire process of the observation of a reference plane can be described as

$$\lambda_{ij}\tilde{\boldsymbol{x}}_{ij} = \begin{bmatrix} A_i & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} {}^{\mathrm{C}}R_i & {}^{\mathrm{C}}\boldsymbol{t}_i \\ \boldsymbol{0}^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} {}^{\mathrm{R}}R_j & {}^{\mathrm{R}}\boldsymbol{t}_j \\ \boldsymbol{0}^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} {}^{\mathrm{P}}R_{ij} & {}^{\mathrm{P}}\boldsymbol{t}_{ij} \\ \boldsymbol{0}^{\mathrm{T}} & 1 \end{bmatrix} \tilde{\boldsymbol{\xi}}_j \tag{13}$$

where $\tilde{\boldsymbol{\xi}}_j$ is a point on the global coordinate system on the reference plane, ${}^{\mathrm{P}}R_{ij}$, ${}^{\mathrm{P}}\boldsymbol{t}_{ij}$ is the transformation between the global and temporal coordinate systems.

Without loss of generality, we can regard the coordinates of the first camera as the global coordinates. In this case, ${}^{\mathrm{C}}R_i$, ${}^{\mathrm{C}}\boldsymbol{t}_i$ are the relative poses from camera 1, and ${}^{\mathrm{R}}R_j = R_{1j}$, ${}^{\mathrm{R}}\boldsymbol{t}_j = \boldsymbol{t}_{1j}$ are the poses of reference planes in the coordinate system of camera 1 (Fig. 3).

3.2 Initial Estimation of the Relative Poses of Cameras and Its Optimization

Placing the first camera at the origin of the global coordinates, the relative poses of cameras and 2-D transformation of the temporal coordinates require only the initial values, where the poses of a reference plane have already been obtained as relative poses in the coordinate system of the first camera. First, we describe the method of estimating the relative pose of each camera using geometric constraints. This part of the process corresponds to relative pose estimation using vanishing points [5].

Let $\boldsymbol{\xi} = [\boldsymbol{r}_1 \ \boldsymbol{r}_2 \ \boldsymbol{t}][X \ Y \ 1]^{\mathrm{T}}$, and then multiplying the cross product of $\boldsymbol{r}_1, \boldsymbol{r}_2$ from the left-hand side yields

$$[(\boldsymbol{r}_1 \times \boldsymbol{r}_2)^{\mathrm{T}}, \ -(\boldsymbol{r}_1 \times \boldsymbol{r}_2) \cdot \boldsymbol{t}]\,\tilde{\boldsymbol{\xi}} = 0. \tag{14}$$

This is a 3-D plane equation and is a pure geometric representation independent of the assigned temporal coordinates. This equation can be reinterpreted using the normal vector $\boldsymbol{n}$ and the distance $-c$ from the origin, and by considering that the pose of a reference plane at observation j is common to all cameras, the estimation of relative camera poses is transformed to the problem of finding R and $\boldsymbol{t}$ for all observed pairs of the normal vector and distance, e.g., $\boldsymbol{n}'_j$ and c'_j, and $\boldsymbol{n}_j$ and c_j, that satisfies

$$\text{find } R, \boldsymbol{t}; \quad \boldsymbol{x}' = R\boldsymbol{x} + \boldsymbol{t} \tag{15}$$

with

$$\begin{cases} \lambda(\boldsymbol{n}_j^{\mathrm{T}}\boldsymbol{x} + c_j) = 0 \\ \lambda'(\boldsymbol{n}_j'^{\mathrm{T}}\boldsymbol{x}' + c'_j) = 0 \end{cases} \tag{16}$$

where $\boldsymbol{n}_j$ and $\boldsymbol{n}'_j$ are the unit normal vectors of the corresponding plane derived from Eq. (14). Substituting Eq. (15) for Eq. (16) and letting $\boldsymbol{x} \leftarrow \boldsymbol{n}_j$, we have

$$\begin{cases} \mu_j = \boldsymbol{n}_j'^{\mathrm{T}} R \boldsymbol{n}_j \\ \boldsymbol{n}_j'^{\mathrm{T}}\boldsymbol{t} = \mu_j c_j - c'_j \end{cases} \tag{17}$$

where $\mu_j = \pm 1$ is a parameter that denotes which side of the j-th plane is observed by camera i. If a reference plane is a single-sided pattern, $\{\mu_j\}$ is always equal to 1. The first equation relates to the rotation and can be solved if $\{\mu_j\}$ is determined by Horn's method [10]:

$$\sum_j \mu_j \boldsymbol{n}_j'^{\mathrm{T}} R(\boldsymbol{q}) \boldsymbol{n}_j \longrightarrow \max \tag{18}$$

where $R(\boldsymbol{q})$ is a rotation matrix represented by a unit quaternion $\boldsymbol{q}$. To compute this rotation, at least three observations are required. If $\{\mu_j\}$ is unknown, maximize $\sum_j (\boldsymbol{n}_j'^{\mathrm{T}} R(\boldsymbol{q}) \boldsymbol{n}_j)^2$. In this case, at least four observations are required. Then, the estimation of translation vector $\boldsymbol{t}$ is carried out using $\{\mu_j\}$.

Next, we compute the initial value of a discrete 2-D transformation in the temporal coordinates. Here, we have the intrinsic parameters A_i, the relative poses of camera ${}^{\mathrm{C}}R_i$ and ${}^{\mathrm{C}}\boldsymbol{t}_i$, and the relative poses of a reference plane ${}^{\mathrm{R}}R_j$ and ${}^{\mathrm{R}}\boldsymbol{t}_j$. Thus, we can directly compute the global coordinates $\boldsymbol{\xi}_j$ from the image coordinates $\boldsymbol{x}_{ij}$. Then, comparing the assigned temporal coordinates $\boldsymbol{\eta}_j$ and global coordinates $\boldsymbol{\xi}_j$, we determine the 2-D rotation ${}^{\mathrm{P}}R_{ij}$ and translation ${}^{\mathrm{P}}\boldsymbol{t}_{ij}$.

$$\boldsymbol{\eta}_j = \frac{1}{\lambda'} H'^{-1} \boldsymbol{x}_{ij} = {}^{\mathrm{P}}R_{ij} \boldsymbol{\xi}_j + {}^{\mathrm{P}}\boldsymbol{t}_{ij} \tag{19}$$

$$H' = A_i [{}^{\mathrm{C}}R_i \ {}^{\mathrm{C}}\boldsymbol{t}_i][{}^{\mathrm{R}}\boldsymbol{r}_{j1} \ {}^{\mathrm{R}}\boldsymbol{r}_{j2} \ {}^{\mathrm{R}}\boldsymbol{t}_j] \tag{20}$$

It is possible to normalize this result as described in 2.2. The following cost function is optimized once, and then we perform optimization again for the final bundle adjustment while fixing the normalized ${}^{\mathrm{P}}R_{ij}$, ${}^{\mathrm{P}}\boldsymbol{t}_{ij}$.

$$\sum_{i,j,k} w_{ijk} ||\boldsymbol{x}_{ijk} - \breve{\boldsymbol{x}}(A_i, \kappa_{i1}, \kappa_{i2}, {}^{\mathrm{C}}R_i, {}^{\mathrm{C}}\boldsymbol{t}_i, {}^{\mathrm{R}}R_j, {}^{\mathrm{R}}\boldsymbol{t}_j, {}^{\mathrm{P}}R_{ij}, {}^{\mathrm{P}}\boldsymbol{t}_{ij}, \boldsymbol{\xi}_k)||^2 \tag{21}$$

We adopt the Levenberg–Marquardt method for this optimization.

3.3 Proposed Method

The proposed method is summarized as follows:

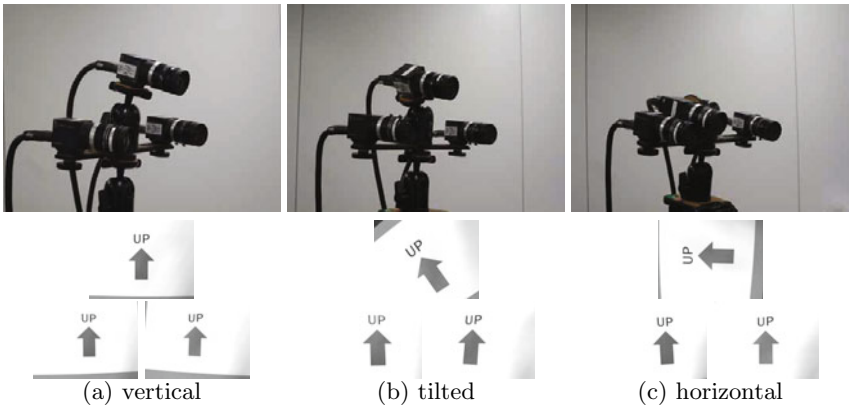
1. Single camera calibration (Zhang's method) for each camera.
2. Estimation of the relative poses of each camera using geometric constraints.
3. Estimation of the correspondence between camera images.
   (a) Computation of 2-D translation by back-projecting image points.
   (b) (Optimization,) Normalization (rounding) of the estimated translation.
4. Bundle adjustment on the corresponding points.

By following these steps, the user can easily perform multi camera calibration by taking a few (≥ 3) images of a reference plane without consideration of poses of each reference plane.

4 Experiments Using Real Images

4.1 Stability Evaluation with a Polka-Dot Pattern

In this section, we evaluate the stability of the proposed method. We used three digital cameras (Point Grey Flea2, VGA, and grayscale) fitted with 6-mm lenses. We performed the proposed method in three configurations: (a) vertical, (b) tilted, and (c) horizontal. We moved only the middle camera (Fig. 4).



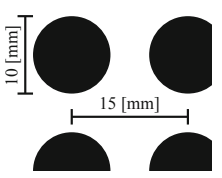
(a) vertical (b) tilted (c) horizontal

Fig. 4. Camera configurations (left: camera 1, right: camera 2, middle: camera 3)

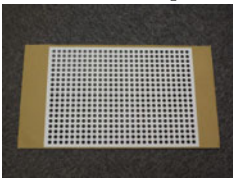
In each configuration, we captured 14 images with the moving reference plane (Fig. 5) and used the center of fitted ellipses[1]. We assigned temporal coordinates as follows. First, we set the center of the circle closest to the image center as the origin and set the directions to the right and toward the bottom of the neighbor circle to $+x$ and $+y$, respectively. Thus, the origins and the basis directions in each image, indicated by the arrows in Fig. 5(c), are different from each other. Figure 5(c) shows the set of images observed for the horizontal case. Here, it is difficult to discern that the image of camera-3 (top center) is rotated by approximately 90 degrees with respect to the other images. Then, we calibrate 1,001 ($= {}_{14}C_4$) quadruples, all of which are combinations chosen from the 14 observations for each case, and calculate the means and standard deviations.

In Table 1, RMS is the root mean square of the reprojection error in pixels. The value of the RMS depends on the camera model, although the average value in the model used herein is within 0.1–0.3. Note that the RMS of Zhang's calibration does not take into account the initial estimate of the relative poses. The RMS of

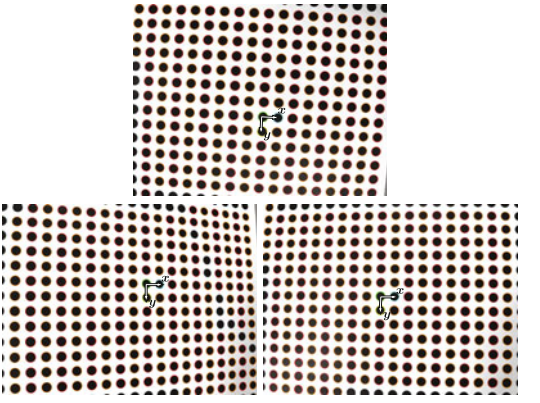
[1] Under perspective projection, since the image of a circle does not form a ellipse, the center of gravity of the fitted ellipse deviates from the image of the center of the circle. This deviation is up to $fr^2/(2D\sqrt{D^2 - r^2})$, where f is the focal length, D is the distance to the center of the circle, and r is the radius of the circle. In the present experiment, this maximum deviation is approx. 0.06 pixels. So this effect is negligible.



(a) Dimensions of the pattern used



(b) Reference plane



(c) Temporal coordinate assignment

Fig. 5. Detection of a polka-dot reference pattern

Table 1. Mean error of 1,001 quadruples in intrinsic parameters ((a) vertical)

	Camera	α	β	γ	u	v	κ_1	κ_2	RMS (SD)
Zhang's calibration (initial value)	1 Mean	825.8	823.6	1.12	326.2	247.0	-0.22	0.15	
	SD	4.84	4.82	0.14	1.93	0.68	0.003	0.012	
	2 Mean	824.4	822.6	1.07	319.5	245.8	-0.22	0.17	
	SD	4.24	4.20	0.20	1.10	0.87	0.003	0.009	
	3 Mean	833.9	832.0	1.31	319.6	253.6	-0.22	0.15	
	SD	4.42	4.49	0.17	0.74	1.31	0.002	0.009	0.151 (0.00329)
After optimization	1 Mean	827.8	825.5	1.13	327.0	247.1	-0.22	0.15	
	SD	3.71	3.74	0.14	1.88	0.67	0.003	0.012	
	2 Mean	825.5	823.7	1.07	319.0	245.9	-0.22	0.17	
	SD	3.28	3.30	0.19	1.28	0.93	0.002	0.009	
	3 Mean	829.7	827.7	1.27	319.3	252.6	-0.21	0.15	
	SD	3.59	3.62	0.16	0.70	1.22	0.002	0.009	0.152 (0.00348)

Zhang's calibration generally provides better results than the proposed method, which has additional constraints.

Table 2 shows that the initial estimates of $\boldsymbol{t}$, $\boldsymbol{q}$, and the optimization results to be similar. This indicates that the initial value estimation of the present study is reasonable. Here, $\boldsymbol{q}$ of the third camera rotates around the $+z$ axis by -40.3 and -87.9 deg, which exactly correspond to configurations (b) and (c), respectively.

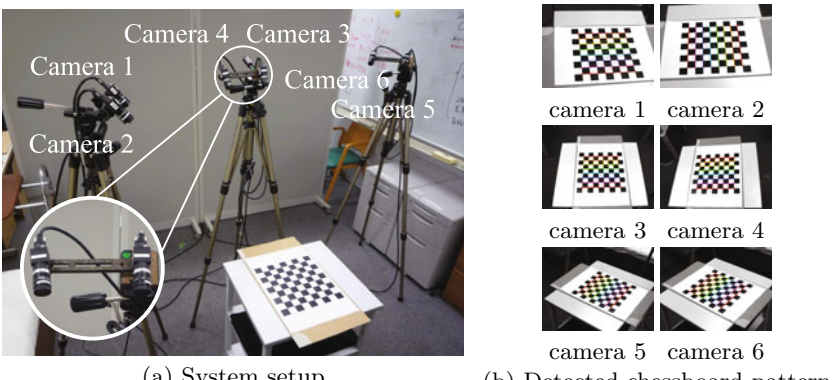
Summarizing these results, even though the proposed method calibrates multiple cameras simultaneously, the proposed method has the same level of stability as Zhang's calibration for a single camera.

4.2 Building a Simple Shape Measurement System

In this section, we arranged six cameras (Flea2), as depicted in Fig. 6(a), and calibrated these cameras. Since the proposed method is independent of the pattern on the reference plane, we adopted a widely used chessboard pattern instead of

Table 2. Difference between initial values and the final results for relative poses

camera	case		Initial value Mean	After optimization Mean	After optimization SD
2	(a)	$\boldsymbol{t}$	(-148.7, 0.4, 33.4)	(-148.8, 0.6, 33.2)	(2.28, 2.74, 1.27)
		$\boldsymbol{q}$	(0.98; 0.00, 0.20, 0.00)	(0.98; 0.00, 0.20, 0.00)	(0.00; 0.00, 0.00, 0.00)
	(b)	$\boldsymbol{t}$	(-150.5, 1.8, 35.4)	(-149.2, 1.5, 34.2)	(2.71, 1.73, 1.21)
		$\boldsymbol{q}$	(0.98; 0.00, 0.20, 0.00)	(0.98; 0.00, 0.20, 0.00)	(0.00; 0.00, 0.00, 0.00)
	(c)	$\boldsymbol{t}$	(-145.8, -2.7, 36.4)	(-147.3, -1.3, 35.4)	(7.92, 6.39, 2.25)
		$\boldsymbol{q}$	(0.98; 0.00, 0.20, 0.00)	(0.98; 0.00, 0.20, 0.00)	(0.00; 0.00, 0.00, 0.00)
3	(a)	$\boldsymbol{t}$	(-73.8, 64.1, 4.5)	(-73.6, 64.8, 2.2)	(1.79, 2.03, 0.88)
		$\boldsymbol{q}$	(0.99; 0.09, 0.10, 0.01)	(0.99; 0.09, 0.09, 0.01)	(0.00; 0.00, 0.00, 0.00)
	(b)	$\boldsymbol{t}$	(-48.5, 106.4, 10.3)	(-47.9, 105.7, 10.2)	(2.24, 2.33, 1.12)
		$\boldsymbol{q}$	(0.94; 0.12, 0.12, -0.30)	(0.94; 0.12, 0.12, -0.30)	(0.00; 0.00, 0.00, 0.00)
	(c)	$\boldsymbol{t}$	(8.6, 119.9, 14.4)	(9.2, 120.5, 14.1)	(5.13, 6.55, 1.60)
		$\boldsymbol{q}$	(0.72; 0.15, 0.10, -0.67)	(0.72; 0.15, 0.10, -0.67)	(0.00; 0.00, 0.00, 0.00)



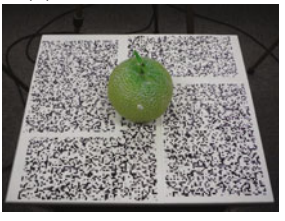
(a) System setup (b) Detected chessboard pattern

Fig. 6. Calibration using a chessboard pattern

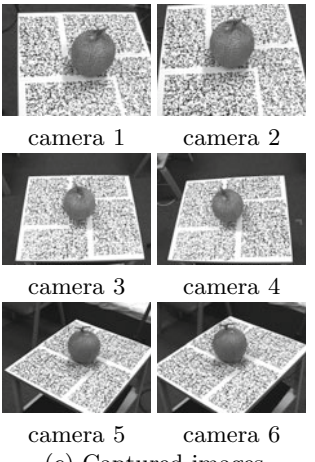
a polka-dot pattern. Points on the chessboard pattern are detected as the point of intersection of two lines, which is more accurate than the detection of a circle. We used OpenCV for the detection (Fig. 6(b)). Figure 6(b) shows that each camera observes the scene in a different direction. Therefore the origins and the basis directions of the assigned temporal coordinates may differ among camera images.

After calibration, we formed camera pairs $(1, 2)$, $(3, 4)$, and $(5, 6)$; and reconstructed the shape of the target using a simple block matching method for each camera pair (Fig. 7(a)). The length of the base line of each pair is approximately 17 [cm], the distance between each camera pair and the target object (a toy melon) is approximately 100 [cm], and the diameter of the target is approximately 13 [cm] (Fig. 7(b)). Figure 7(c) shows the images captured by each camera. The reconstruction results for each pair are shown in Figs. 8(a) through 8(c). Figures 8(d) and 8(e) show the results of displaying all of the reconstructed

(a) Observation scene



(b) Target object



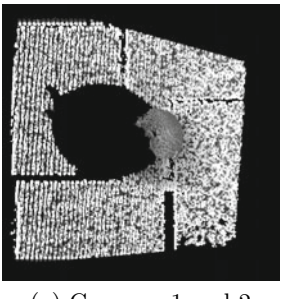
camera 1 camera 2

camera 3 camera 4

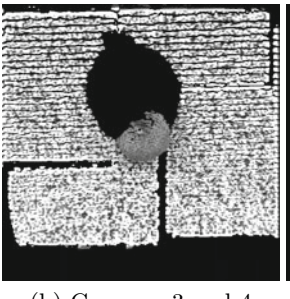
camera 5 camera 6

(c) Captured images

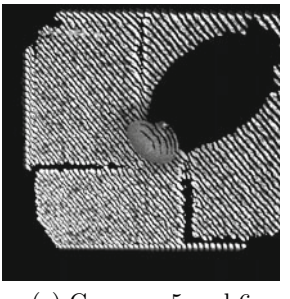
Fig. 7. 3-D shape measurement



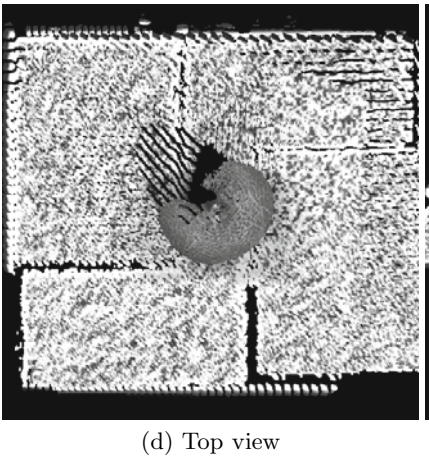
(a) Cameras 1 and 2



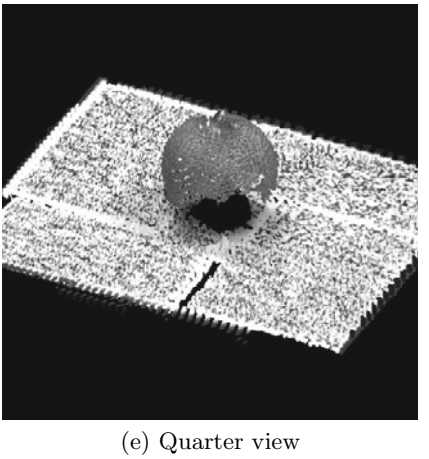
(b) Cameras 3 and 4



(c) Cameras 5 and 6



(d) Top view



(e) Quarter view

Fig. 8. 3-D reconstruction result

results together from different viewpoints. We confirmed that the object shape obtained using the proposed method is reasonable and that the relative poses of each camera are correctly estimated. Therefore, we believe that the proposed calibration method has sufficient accuracy for practical application.

5 Conclusion

In the present paper, we propose a calibration method for multi camera systems using a simple patterned reference plane. The proposed method can be used to perform full calibration for each camera and estimate the relative poses of the cameras without the requirement for correspondence points. Furthermore, the proposed method can optimize parameters in the sense of bundle adjustment by estimating the 2-D discrete transformation of an observed pattern on a reference plane. The proposed method is stable in practice. Therefore the user can easily construct a simple full-calibrated 3-D shape measurement system.

References

1. Zhang, Z.: A flexible new technique for camera calibration. tPAMI 22, 1330–1334 (2000)
2. Tsai, R.: A versatile camera calibration techniaue for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. IEEE J. Robot. and Autom. 3, 323–344 (1987)
3. Pollefeys, M., Koch, R., Gool, L.V.: Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. IJCV 32, 7–25 (1999)
4. Sturm, P.: Algorithms for plane-based pose estimation. In: Proc. CVPR 2000, Hilton Head Island, South Carolina, USA, pp. 1010–1017 (2000)
5. Caprile, B., Torre, V.: Using vanishing points for camera calibration. IJCV 4, 127–139 (1990)
6. Ueshiba, T., Tomita, F.: Plane-based calibration algorithm for multi-camera systems via factorization of homography matrices. In: Proc. ICCV 2003, pp. 966–973 (2003)
7. Ramalingam, S., Sturm, P., Lodha, S.K.: Towards complete generic camera calibration. In: Proc. CVPR 2005, vol. 1, pp. 1093–1098 (2005)
8. Fiala, M.: ARTag, a fiducial marker system using digital techniques. In: Proc. CVPR 2005, vol. 2, pp. 590–596 (2005)
9. Lowe, D.: Object recognition from local scale-invariant features. In: Proc. ICCV 1999, pp. 1150–1157 (1999)
10. Horn, B.: Closed-form solution of absolute orientation using unit quaternions. J. the Optical Society of America A 4, 629–642 (1987)

Over-Segmentation Based Background Modeling and Foreground Detection with Shadow Removal by Using Hierarchical MRFs

Te-Feng Su, Yi-Ling Chen, and Shang-Hong Lai

Department of Computer Science, National Tsing Hua University,
No. 101, Section 2, Kuang-Fu Road, Hsinchu, Taiwan 30013, R.O.C.

Abstract. In this paper, we propose a novel over-segmentation based method for the detection of foreground objects from a surveillance video by integrating techniques of background modeling and Markov Random Fields classification. Firstly, we introduce a fast affinity propagation clustering algorithm to produce the over-segmentation of a reference image by taking into account color difference and spatial relationship between pixels. A background model is learned by using Gaussian Mixture Models with color features of the segments to represent the time-varying background scene. Next, each segment is treated as a node in a Markov Random Field and assigned a state of foreground, shadow and background, which is determined by using hierarchical belief propagation. The relationship between neighboring regions is also considered to ensure spatial coherence of segments. Finally, we demonstrate experimental results on several image sequences to show the effectiveness and robustness of the proposed method.

1 Introduction

Extracting foreground objects from image sequences is a critical task for many computer vision applications, such as video processing, visual surveillance and object recognition. Background subtraction is a core component for video surveillance, whose objective is to discriminate the foreground from the background scene. To achieve this goal, a robust background modeling technique is essential. The basic idea of background modeling is to maintain an estimation of the background image model which represents the scene with no foreground objects. Then, moving objects can be detected by a simple subtraction and thresholding procedure. Hence, the more accurate the background model, the more accurate is the detection of the foreground objects.

Most traditional background modeling techniques are pixel-based, and they usually estimate the probability of the individual pixels belonging to background by using GMMs [1] or to label each pixel as foreground or background by MRFs [2]. However, pixel-based models are less efficient and effective in handling illumination change and dynamic scene such as swaying vegetation, waving trees, fluttering flags, and so on. Even though the background is static, camera

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 535–546, 2011.

jittering and signal noise may still cause non-stationary problems. Several block-based methods were developed to overcome such problems, which partition a background image into sub-blocks to utilize block correlation [3] or to compute block-specific features, such as *local binary pattern* histograms [4]. However, the fix-sized blocks often fail to correctly classify the foreground objects because they are not well fitted to object boundaries, which also results in inaccurate shape of the detected foreground objects.

Motivated by the above issues, we propose a novel over-segmentation based approach for foreground objects detection in this paper. Unlike pixel-based or block-based methods, the proposed method exploits the observation that neighboring pixels are very likely to have the same foreground or background classification if they are appropriately grouped together according to certain similarity measure. Despite the simplicity of dividing an image into blocks, the subdivided regions usually do not fit to the object boundaries well. We thus propose a fast and effective affinity propagation algorithm to obtain the over-segmentation of a background image to facilitate the task of background modeling. By considering color and spatial coherence of neighboring pixels, the proposed method is capable of handling illumination change in a scene effectively. In the following

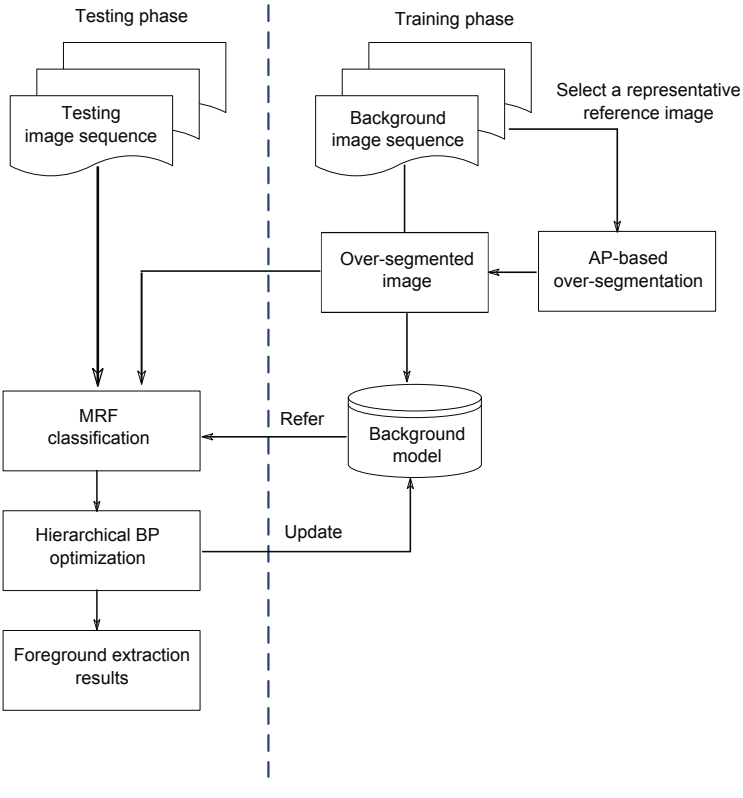


Fig. 1. System flowchart

foreground/background classification stage, we produce the over-segmentation of various resolutions on the reference image to form a hierarchy of MRFs and each segment is treated as a node. Hierarchical belief propagation [5] is then utilized to label the MRFs.

Fig. 1 illustrates the flowchart of the proposed background modeling and foreground detection system. In the training phase, an reference background image is selected and its over-segmentation is produced by performing fast AP clustering algorithm. Based on the over-segmented image, the background models are learned via GMMs and a hierarchy of MRFs is constructed. Foreground object detection is accomplished through MRF classification by hierarchical belief propagation when new images are acquired from a test sequence and the corresponding background models are updated accordingly.

The rest of this paper is organized as follows. In Section 2, we first briefly review the related work. Then, the over-segmentation based background modeling method is introduced in Section 3. In Section 4, the detection of foreground objects by MRFs classification is described. In Section 5, we show some experimental results and quantitative comparisons to demonstrate the superior performance of the proposed method over previous methods. Section 6 concludes this paper.

2 Related Work

Many approaches for background subtraction have been proposed over the past decades, but they usually differ in the ways of modeling the background. Most of them can be classified as pixel-based approaches. A well-known method by Grimson and Stauffer [1] proposed to use GMMs for background modeling. It describes each pixel as a mixture of Gaussians and updates the models adaptively according to the input image sequence. Zivkovic proposed an improved GMM learning algorithm that estimates the parameters of the GMM and simultaneously selects the number of Gaussians [6]. Elgammal et al. introduced nonparametric estimation method for per-pixel background modeling [7]. They utilized a general nonparametric kernel density estimation technique for building a statistical representation of the background scene.

Spatial and temporal neighboring relationships of pixels are useful information for object segmentation. In [2], Paragios and Ramesh proposed a MRF-based method to deal with change detection for subway monitoring. Migdal and Grimson adopted MRFs to model spatial and temporal relationship of neighborhood pixels [8]. Wang et al. [9] introduced dynamic CRFs for foreground object and moving shadow segmentation in indoor video scene. An approximate filtering algorithm is exploited to update parameters of CRF models according to previous frames. Huang et al. [10] proposed a region-based motion segmentation algorithm to obtain motion-coherence regions. Both spatial and temporal coherence of regions are taken into account to maintain the continuity of segmentation by using MRFs.

Recently, several block-based methods were developed for background modeling and subtraction to more effectively deal with illumination change and

dynamic scenes. Generally, block-based algorithms start by dividing a background image into blocks and construct the background models by calculating block-specific features extracted from these blocks. In [3], the correlation between blocks is measured by the *normalized vector distance* to realize robust background subtraction against varying illumination. Heikkila and Pietikinen [4] proposed to model the background scene based on Local Binary Pattern (LBP) histogram and produce coarse detection of foreground object. However, the LBP histogram cannot capture temporal variation in the pattern. Following [4], Chen et al. proposed a contrast histogram measure to describe each block and performed object detection by combining a pixel-level GMMs and block-wise contrast descriptors [11].

Cast shadows are difficult to be correctly detected by most background subtraction methods. It is often misclassified as the foreground region, resulting in inaccurate object shapes and the degradation of model updating. Shadow detection techniques can be classified into two groups: model-based and property-based techniques. Model-based techniques rely on models representing the prior knowledge of the geometry of the scene or objects, and the illumination [12]. Property-based techniques identify shadows by using features, such as brightness [13,14], geometry [15] or texture [16].

3 Over-Segmentation Based Background Modeling

To enable the background model to more effectively handle changes occurred in the scene, it is preferable to divide a background scene into sub-regions and learn the background models accordingly. To this end, we propose a simple and efficient affinity propagation clustering algorithm to over-segment a reference image I_R among an input sequence before the background model learning process. Furthermore, a hierarchy of over-segmentation built over I_R is constructed to facilitate the following foreground object detection by MRF classification (Section 4). In this work, we adopt GMMs [1] as underlying the background models, which are trained for both pixel and segmentation level.

Affinity propagation (AP) [17] is an iterative algorithm that groups data points into clusters by sending *messages* between data points. The pairwise *similarity* $s(i,k)$, which measures how well-suited data point k is to be the *exemplar* (i.e. cluster center) for data point i, is taken as input and AP aims to search for a number of clusters such that the net similarity is maximized. Unlike other clustering techniques, such as the k-means clustering that needs the number of clusters to be explicitly specified, AP takes for each data point k a *preference* value $s(k,k)$ as input that indicates a candidate exemplar's potential of being chosen as an exemplar. Exemplars emerge during the process of message passing and the number of identified exemplars depends on the input preference values. There are two kinds of messages to be updated during each iteration, i.e. *responsibility* and *availability*, and each accounts for a different kind of competition. Briefly speaking, responsibility update lets all candidate exemplars compete for ownership of a data point while availability update collects evidence from data

Algorithm 1. Affinity propagation

Initialization:

$r(i,k) = 0,\ a(k,i) = 0$ for all $i,\ k$

Responsibility updating:

$r(i,k) \leftarrow s(i,k) - \max_{j:j\neq k}\{a(j,i) + s(i,j)\}$

Availability updating:

$a(k,k) \leftarrow \sum_{j:j\neq k} \max\{0,\ r(j,k)\}$

$a(k,i) \leftarrow \min\{0,\ r(k,k) + \sum_{j:j\notin\{k,i\}} \max\{0,\ r(j,k)\}\}$

Exemplar assignments:

$c_i^* \leftarrow \arg\max_k\ r(i,k) + a(k,i)$

points reflecting the competence of each candidate exemplar. The message updating procedures are summarized in Algorithm 1.

The messages are directional. The responsibility $r(i,k)$, sent from data point i to candidate exemplar k, delivers the accumulated evidence for how well-suited it is to assign point i to point k, by considering other potential exemplars' competition for point i. The availability $a(k,i)$, sent from candidate exemplar k to point i, delivers the accumulated evidence for how appropriate it would be for point i to choose point k as its exemplar, by considering the aggregate support of choosing point k to be an exemplar from other points. After convergence, availabilities and responsibilities can be combined to identify exemplars. For point i, it is assigned to the exemplar c_i that maximizes $r(i,k) + a(k,i)$.

In the application of image over-segmentation, neighboring pixels of similar color are grouped together and each pixel competes for exemplarship by exchanging messages with each other. One drawback of AP clustering is its high complexity of message updating, which is $O(N^2)$ if each pixel sends messages to all the other pixels. To achieve efficiency, we exploit the assumption that distant pixels are not possible to be assigned to the same exemplar and thus message exchange between pixels far away is not necessary. To further reduce the amount of messages, we take advantage of a set of *virtual* exemplars, which are responsible for competing for pixels, to form the over-segmentation. We do not define pixel-to-pixel similarity since the image pixels are no longer candidates of exemplars to form the final segmentation. Therefore, the amount of messages to be updated is greatly reduced, leading to an efficient algorithm.

Following the convention of AP, we define the following negative real-valued similarity measure between a pixel p and its nearby virtual exemplar v, taking into account color difference and spatial relationship

$$s(p,v) = -(\lambda_c\|\mathbf{c}_p - \mathbf{c}_v\|^2 + \lambda_s\|\mathbf{u}_p - \mathbf{u}_v\|^2), \tag{1}$$

where λ_c and λ_s are weighting coefficients to balance the two terms. Initially, we obtain an initial over-segmentation by partitioning the input image into a regular grid consisting of fix-sized blocks, e.g. 8×8, and associate each block

with a virtual exemplar. The mean color $\mathbf{c}_v$ and position $\mathbf{u}_v$ of each block are then computed for the associated v for message computation. Each pixel only sends messages to the virtual exemplars with the corresponding segments connected to each other. The message updating procedure of AP remains unchanged. After each iteration of AP clustering, the pixels assigned to each segment vary and the exemplar attributes $\mathbf{u}_v$ and $\mathbf{c}_v$ are updated accordingly. Generally, 5∼10 iterations suffice to obtain good segmentation results in our experiments.

4 Markov Random Fields for Classification

4.1 Energy Minimization

In this paper, we formulate the foreground/background classification problem as labeling a MRF with each node corresponding to a pixel or segment in an image. Let $\mathcal{S} = \{s_1, s_2, ..., s_n\}$ be the set of nodes in a graph $\mathcal{G}$ and $\mathcal{E}$ be the set of edges with $(s_i, s_j) \in \mathcal{E}$ indicating that there is an edge connecting s_i and s_j. We aim to find an optimal *configuration* $\widehat{\Omega}$ of $\mathcal{G}$ that assigns a label $l_i \in \{foreground, background, shadow\}$ for node s_i such that the following energy function is minimized:

$$E(\Omega) = \sum_{s_i \in \mathcal{S}} V_{likelihood}(l_i) + \alpha \sum_{(s_i,s_j)\in\mathcal{E}} V_{prior}(l_i, l_j), \tag{2}$$

where α is a weighting coefficient. The energy function $E(\Omega)$ is the sum of two terms: *likelihood energy* $V_{likelihood}$ and *prior energy* V_{prior}. The likelihood term $V_{likelihood}$ measures the likelihood that a node s_i is classified as one of its three possible states and is composed of the weighted combination of two terms: *color distortion* V^C and *gain* information V^G:

$$V_{likelihood}(l_i) = \lambda \sum_{s_i \in \mathcal{S}} V^C(l_i) + (1-\lambda) \sum_{s_i \in \mathcal{S}} V^G(l_i). \tag{3}$$

Color distortion V^C is the angle between the color vectors associated with a node s_i in the current observed image and the corresponding background model. Note that if s_i corresponds to a segment, the average color vector is used to compute V^C to measure the similarity with its corresponding background model. The *gain* information V^G was designed to handle cast shadow based on the observation that shadow regions are expected to possess lower luminance but similar chromaticity values [14]. It is calculated by the ratio between the brightness change over the corresponding background model,

$$gain = \frac{I_o - I_b}{I_b} \tag{4}$$

where I_b and I_o are the average intensity of background model and the observed region, respectively. The variation of intensity in the shadow regions due to illumination changes should be relatively small.

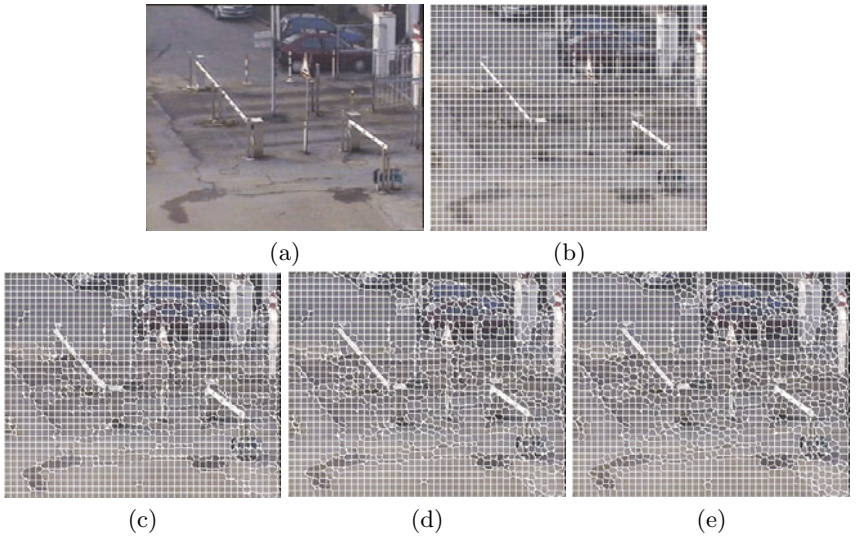


Fig. 2. Over-segmenting a reference image from 8×8 regular grid by our fast AP clustering algorithm. (a) original image (b) 8×8 regular grid image (c) AP based clustering algorithm with 1 iteration (d) AP-based clustering algorithm with 3 iterations (e)final over-segmentation image from AP-based clustering algorithm.

Let R_i and R_i^b be the colors corresponding to the ith pixels and mean color for segment case in an observed image F_o and background image F_b, respectively. V^C and V^G terms are defined as follows:

$$V_{(l_i)}^C = \begin{cases} 1 - exp(-f_{cd}(R_i, R_i^b)) \; if \; l_i = background \\ exp(-f_{cd}(R_i, R_i^b)) \; otherwise \end{cases} \tag{5}$$

$$V_{(l_i)}^G = \begin{cases} 1 - exp(-f_{gain}(R_i, R_i^b)) \; if \; l_i = background, shadow \\ exp(-f_{gain}(R_i, R_i^b)) \; otherwise \end{cases} \tag{6}$$

where f_{cd} and f_{gain} are functions to calculate color distortion and gain information between R_i and R_i^b, respectively. The definition for f_{cd} and f_{gain} can be formed by:

$$f_{cd}(m, n) = \arccos(\frac{\overrightarrow{m} \cdot \overrightarrow{n}}{|\overrightarrow{m}| \; |\overrightarrow{n}|}) \tag{7}$$

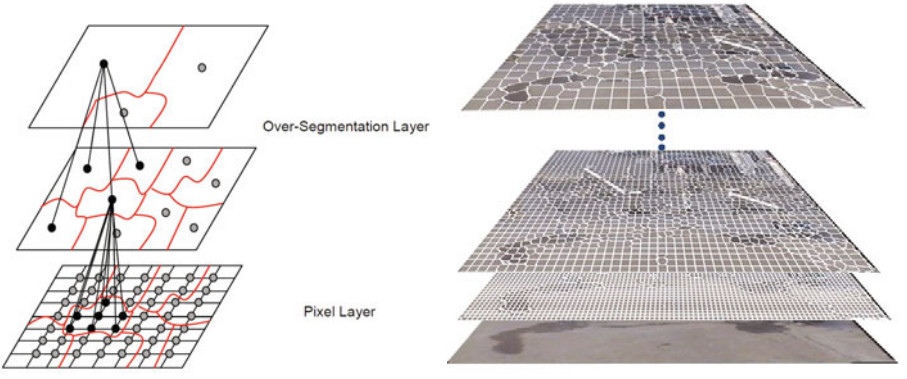
$$f_{gain}(I_o, I_b) = \frac{I_o - I_b}{I_b} \tag{8}$$

where m and n are two input of color vectors.

The prior energy V_{prior} captures the spatial continuity between neighboring pixels or segments. It introduces more penalty if two neighboring regions with small color distortion are assigned different labels. Let R_i and R_j be the pixels

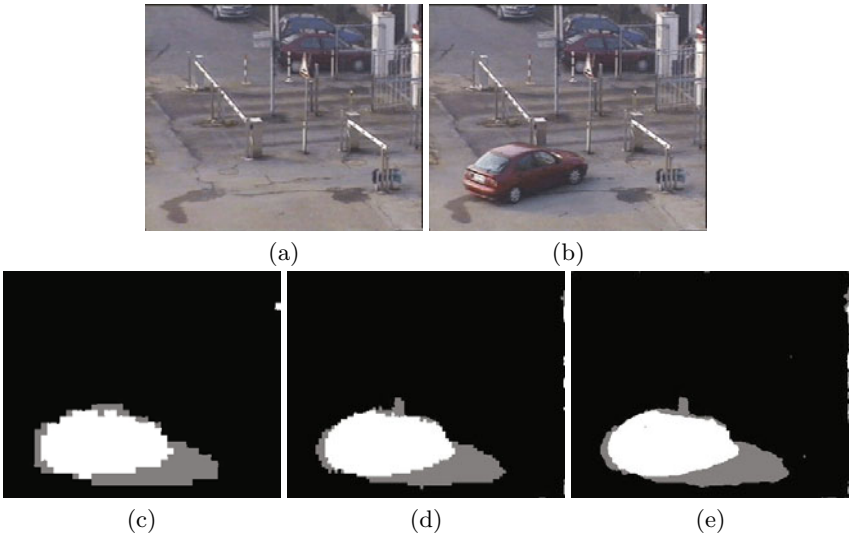
or segments corresponding two connected nodes s_i and s_j in $\mathcal{G}$. We thus define V_{prior} as follows,

$$V_{prior} = \begin{cases} 1 - \exp(-f_{cd}(R_i, R_j)), & if \quad l_i \neq l_j \\ 0, & if \quad l_i = l_j. \end{cases} \tag{9}$$



(a) (b)

Fig. 3. (a) Hierarchical MRFs built over pixel and segmentation levels. (b) Example of hierarchical over-segmentation.



(a) (b)

(c) (d) (e)

Fig. 4. A coarse to fine foreground and shadow detection results by our proposed method from "campus" image sequence. (a) original image (b) frame 65 of "campus" image sequence. (c)∼(e) show our foreground and shadow detection results from coarse to fine.

4.2 Hierarchical Belief Propagation Optimization over MRFs

After the MRF is built, belief propagation (BP) algorithm is employed to find the optimal label assignment of each node by minimizing the energy function defined in Equation (2). In this work, we build a hierarchy of MRFs on both pixel and segmentation levels. By using the proposed fast AP clustering algorithm, we over-segment the reference background image with various initial block sizes. As shown in Fig. 3, a segment in the over-segmented image is viewed as a node in the MRF model. A segment in a coarser level is obtained by merging some neighboring segments in the next finer level. Therefore, we can easily build a hierarchical MRF structure from finest level to coarsest level as shown in Fig. 3.

We exploit the hierarchical BP algorithm [5] to solve the optimization problem defined in the last subsection. The messages at the coarsest level are initialized as zero and the messages after convergence at each level are passed to the successive finer level as the initial guesses for BP message updating. Fig. 4 shows an example of coarse-to-fine foreground and shadow detection by hierarchical MRF classification.

5 Experimental Results

To evaluate the performance of the proposed over-segmentation based background subtraction, three image sequences from public domain are adopted as benchmark.

The three sequences are "*campus*", "*intelligent room*" and "*hall monitoring*", which are taken from various types of scenes, such as outdoor and indoor environments, to demonstrate the robustness of the proposed method. Fig. 6 compares the results by the proposed method to those by [6] and [14]. Obviously,

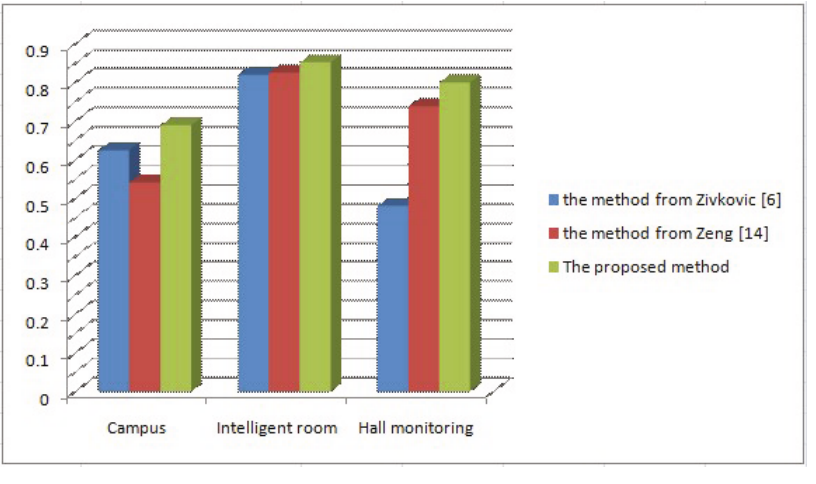
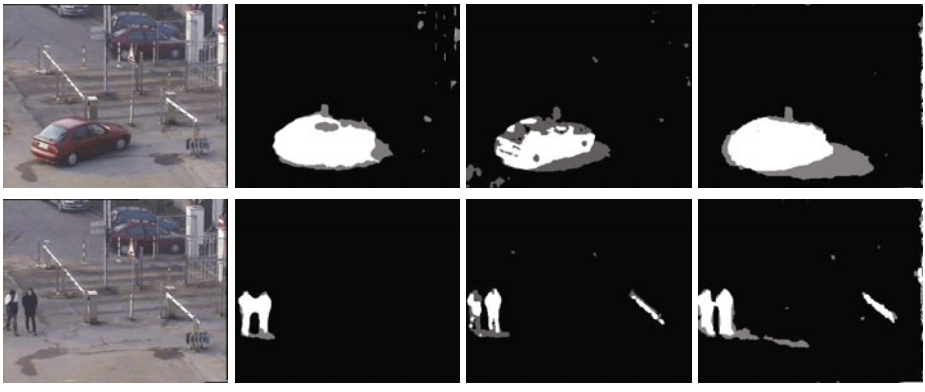
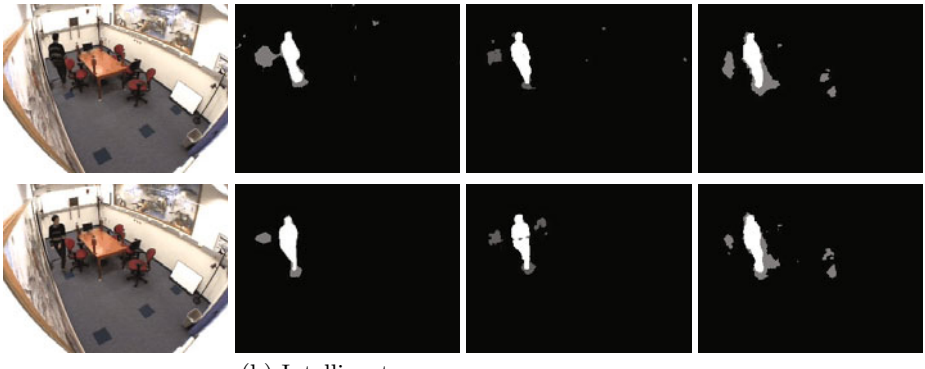


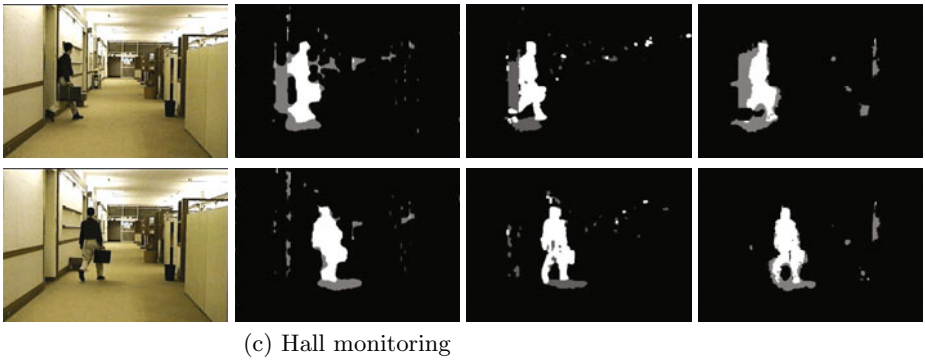
Fig. 5. Quantitative comparison between different methods in the three static scene sequences



(a) Campus



(b) Intelligent room



(c) Hall monitoring

Fig. 6. Some background subtraction results of different methods. The white and gray pixels indicate the detected foreground and shadow regions. First column: original image. Second column: the results by Zivkovic [6]. Third column: the results by Zeng [14]. The right-most column: the results by the proposed method.

the foreground objects, such as the moving car and pedestrians in the *campus* sequence, detected by our method is more accurate than previous methods. In *hall monitoring* sequence, the proposed method is more robust against the noise due to light fluctuation. Cast shadows caused by the foreground objects are also detected well in all test sequences.

To provide quantitative evaluation, we use the similarity measure presented by Li et al. [18] in this paper. Let A be a detected region and B be the corresponding ground truth. The similarity measure between A and B is defined as

$$S(A,B) = \frac{A \bigcap B}{A \bigcup B} \tag{10}$$

$S(A,B)$ reaches its maximal value of 1 if A and B is exactly the same. Otherwise, $S(A,B)$ fluctuates between 0 to 1 depending on their overlapped regions. The ground truth data are obtained from public domain and the residuals are marked manually. The quantitative comparisons shown in Fig. 5 indicates the superior performance of the proposed method over the previous methods.

6 Conclusion

In this paper, a new over-segmentation based background modeling algorithm is presented for foreground and shadow segmentation. The proposed method uses a fast AP clustering algorithm to obtain image over-segmentation of various resolutions. The foreground/background classification is then formulated as an energy minimization problem over the MRFs constructed on the segmented image by using hierarchical belief propagation. Experimental results on several test sequences and quantitative analysis show that the proposed method performs well for foreground object extraction and cast shadow detection.

Acknowledgement. This work was partially supported by National Science Council in Taiwan under the project 98-2220-E-007-029.

References

1. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: IEEE Conference on CVPR, vol. 2, p. 252 (1999)
2. Paragios, N., Ramesh, V.: A mrf-based approach for real-time subway monitoring. In: IEEE Conference on CVPR, vol. 1, pp. I–1034–I–1040 (2001)
3. Matsuyama, T., Ohya, T., Habe, H.: Background subtraction for nonstationary scenes. In: Proceedings of ACCV, pp. 662–667 (2000)
4. Heikkila, M., Pietikainen, M.: A texture-based method for modeling the background and detecting moving objects. IEEE Transactions on PAMI 28, 657–662 (2006)
5. Felzenszwalb, P., Huttenlocher, D.: Efficient belief propagation for early vision. International Journal of Computer Vision 70, 41–54 (2006)
6. Zivkovic, Z., van der Heijden, F.: Recursive unsupervised learning of finite mixture models. IEEE Transactions on PAMI 26, 651–656 (2004)

7. Elgammal, A., Harwood, D., Davis, L.: Non-parametric model for background subtraction. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1843, pp. 751–767. Springer, Heidelberg (2000)
8. Migdal, J., Grimson, W.E.L.: Background subtraction using markov thresholds. In: IEEE Workshop on Motion and Video Computing, vol. 2, pp. 58–65 (2005)
9. Wang, Y., Loe, K.F., Wu, J.K.: A dynamic conditional random field model for foreground and shadow segmentation. IEEE Transactions on PAMI 28, 279–289 (2006)
10. Huang, S.S., Fu, L.C., Hsiao, P.Y.: Region-level motion-based background modeling and subtraction using mrfs. IEEE Transactions on IP 16, 1446–1456 (2007)
11. Chen, Y.T., Chen, C.S., Huang, C.R., Hung, Y.P.: Efficient hierarchical method for background subtraction. Pattern Recognition 40, 2706–2715 (2007)
12. Martel-Brisson, N., Zaccarin, A.: Learning and removing cast shadows through a multidistribution approach. IEEE Transactions on PAMI 29, 1133–1146 (2007)
13. Cucchiara, R., Grana, C., Piccardi, M., Prati, A.: Detecting moving objects, ghosts, and shadows in video streams. IEEE Transactions on PAMI 25, 1337–1342 (2003)
14. Zeng, H.C., Lai, S.H.: Adaptive foreground object extraction for real-time video surveillance with lighting variations. In: ICASSP, pp. I–1201–I–1204 (2007)
15. Salvador, E., Cavallaro, A., Ebrahimi, T.: Cast shadow segmentation using invariant color features. Computer Visual Image Understand 95, 238–259 (2004)
16. Zhang, W., Fang, X.Z., Yang, X., Wu, Q.: Moving cast shadows detection using ratio edge. IEEE Transactions on Multimedia 9, 1202–1214 (2007)
17. Frey, B.J., Dueck, D.: Clustering by passing messages between data points. Science 315, 972–976 (2007)
18. Li, L., Huang, W., Gu, I.Y.H., Tian, Q.: Statistical modeling of complex backgrounds for foreground object detection. IEEE Transactions on IP 13, 1459–1472 (2004)

MRF-Based Background Initialisation for Improved Foreground Detection in Cluttered Surveillance Videos

Vikas Reddy, Conrad Sanderson, Andres Sanin, and Brian C. Lovell

NICTA, PO Box 6020, St Lucia, QLD 4067, Australia
The University of Queensland, School of ITEE, QLD 4072, Australia

Abstract. Robust foreground object segmentation via background modelling is a difficult problem in cluttered environments, where obtaining a clear view of the background to model is almost impossible. In this paper, we propose a method capable of robustly estimating the background and detecting regions of interest in such environments. In particular, we propose to extend the background initialisation component of a recent patch-based foreground detection algorithm with an elaborate technique based on Markov Random Fields, where the optimal labelling solution is computed using iterated conditional modes. Rather than relying purely on local temporal statistics, the proposed technique takes into account the spatial continuity of the entire background. Experiments with several tracking algorithms on the CAVIAR dataset indicate that the proposed method leads to considerable improvements in object tracking accuracy, when compared to methods based on Gaussian mixture models and feature histograms.

1 Introduction

One of the low-level tasks in most intelligent video surveillance applications (such as person tracking and identification) is to segment objects of interest from an image sequence. Typical segmentation approaches employ the idea of comparing each frame against a model of the background, followed by selecting the outliers (i.e., pixels or areas that do not fit the model). However, most methods presume the training image sequence used to model the background is free from foreground objects. This assumption is often not true in the case of uncontrolled environments such as train stations and motorways, where directly obtaining a clear background is almost impossible. Furthermore, in outdoor video surveillance a strong illumination change can render the existing background model ineffective (e.g., due to introduction of shadows), thereby forcing us to compute a new background model. In such circumstances, it becomes inevitable to reinitialise the background model using cluttered sequences (i.e., where parts of the background are occluded). Robust background initialisation in these scenarios can result in improved segmentation of foreground objects, which in turn can lead to more accurate tracking.

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 547–559, 2011.

The majority of the algorithms described in the literature, such as [1,2,3,4], do not have a robust strategy to handle cluttered sequences. Specifically, they fail when the background in the training sequence is exposed for a shorter duration than foreground objects. This is due to the model being initialised by relying solely on the temporal statistics of the image data, which is easily affected by the inclusion of foreground objects in the training sequence.

To alleviate this problem, a few algorithms have been proposed to initialise the background image from cluttered image sequences. Typical examples include median filtering, finding pixel intervals of stable intensity in the image sequence [5], building a codebook for the background model [3], agglomerative clustering [6] and minimising an energy function using an α–*expansion* algorithm [7]. However, none of them evaluate the foreground segmentation accuracy using their estimated background model.

In this paper, we propose to replace the background model initialisation component of a recently introduced foreground segmentation method [1] and show that the performance can be considerably improved in cluttered environments. The proposed background initialisation is carried out in a Markov Random Field (MRF) framework, where the optimal labelling solution is computed using iterated conditional modes. The spatial continuity of the background is also considered in addition to the temporal statistics of the training sequence. This strategy is particularly robust to training sequences containing foreground objects exposed for longer duration than the background over a given time interval.

Experiments on the CAVIAR dataset, where most of the sequences contain occluded backgrounds, show that the proposed framework (MRF + multi-stage classifier) yields considerably better results in terms of tracking accuracy than the baseline multi-stage classifier method [1] as well as methods based on Gaussian mixture models [8] and feature histograms [9].

We continue as follows. The overall foreground segmentation framework is described in Section 2, followed by the details of the proposed MRF-based background initialisation method in Section 3. Performance evaluations and comparisons with three other algorithms are given in Section 4, followed by the main findings in Section 5.

2 Foreground Segmentation Framework

We build on the patch-based multi-stage foreground segmentation method proposed in [1], which has four major components:

1. Division of a given image into overlapping blocks (patches), followed by generating a low-dimensional 2D Discrete Cosine Transform (DCT) based descriptor for each block [10].
2. Classification of each block into foreground or background based on a background model, where each block is sequentially processed by up to three classifiers. As soon as one of the classifiers deems that the block is part of the background, the remaining classifiers are not consulted. In sequential order of processing, the three classifiers are:

   (a) a probability measurement according to a location specific multivariate Gaussian model of the background (i.e., one Gaussian for each block location);
   (b) an illumination robust similarity measurement through a cosine distance metric;
   (c) a temporal correlation check where blocks and decisions from the previous image are taken into account.
3. Model reinitialisation to address scenarios where a sudden and significant scene change can make the current model inaccurate.
4. Probabilistic generation of the foreground mask, where the classification decisions for all blocks are integrated. The overlapping nature of the analysis is exploited to produce smooth contours and to minimise the number of errors (both false positives and false negatives).

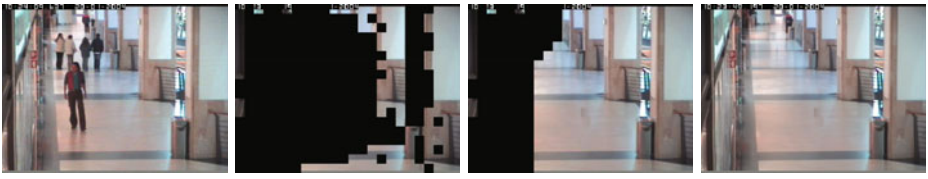
Parts 2(a) and 2(b) require a location specific Gaussian model, which can be characterised by a mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. In an attempt to allow the training sequence to contain moving foreground objects, a rudimentary Gaussian selection strategy is employed in [1]. Specifically, for each block location a two-component Gaussian mixture model (GMM) is trained, followed by taking the absolute difference of the weights of the two Gaussians. If the difference is greater than 0.5, the Gaussian with the dominant weight is retained. The reasoning is that the less prominent Gaussian is modelling moving foreground objects and/or other outliers. If the difference is less than 0.5, it is assumed that no foreground objects are present and all available data for that particular block location is used to estimate the parameters of the single Gaussian.

There are several problems with the above parameter selection approach. It is assumed that foreground objects are either continuously moving in the sequence or that no object stays in one location for more than 25% of the length of the training sequence. This is not guaranteed to occur in uncontrolled environments such as railway stations. The decision to retain the dominant Gaussian solely relies on local temporal statistics and ignores rich local spatial correlations that naturally exist within a scene.

To address the above problems, we propose to estimate the parameters of the background model via a Markov Random Field (MRF) framework, where in addition to temporal information, spatial continuity of the entire background is considered. The details of the MRF-based algorithm are given in the following section.

3 Proposed Background Initialisation Algorithm

Let the resolution of the image sequence I be $\mathcal{W} \times \mathcal{H}$, with ϕ colour channels. The proposed algorithm has three main stages: **(1)** division of each frame into non-overlapping blocks and collection of possible background blocks over a given time interval, **(2)** partial background reconstruction using unambiguous blocks, **(3)** ambiguity resolution through exploitation of spatial correlations



(i) **(ii)** **(iii)** **(iv)**

Fig. 1. Example of background estimation from an image sequence cluttered with foreground objects: **(i)** example frame, **(ii)** partial background initialisation (after stage 2), **(iii)** remaining background estimation in progress (stage 3), **(iv)** estimated background

across neighbouring blocks. An example of the algorithm in action is shown in Fig. 1. The details of the three stages are given below.

In stage 1, each frame is viewed as an instance of an undirected graph, where the nodes of the graph are blocks of size $N \times N \times \phi$ pixels[1]. We denote the nodes of the graph by $\mathcal{N}(i,j)$ for $i = 0, 1, 2, \cdots, (\mathcal{W}/N) - 1$, $j = 0, 1, 2, \cdots, (\mathcal{H}/N) - 1$. Let I_f be the f-th frame of the training image sequence and let its corresponding node labels be denoted by $\mathcal{L}_f(i,j)$, and $f = 1, 2, \cdots, F$, where F is the total number of frames. For convenience, each node label $\mathcal{L}_f(i,j)$ is vectorised into an ϕN^2 dimensional vector $\mathbf{l}_f(i,j)$. In comparison to pixel-based processing, block-based processing is more robust against noise and captures better contextual spatial continuity of the background.

At each node (i,j), a representative set $\mathcal{R}(i,j)$ is maintained. It contains only unique representative labels, $\mathbf{r}_k(i,j)$ for $k = 1, 2, \cdots, S$ (with $S \leq F$) that were obtained along its temporal line. To determine uniqueness, the similarity of labels is calculated as described in Section 3.1. Let weight W_k denote the number of occurrences of $\mathbf{r}_k$ in the sequence, i.e., the number of labels at location (i,j) which are deemed to be the same as $\mathbf{r}_k(i,j)$.

It is assumed that one element of $\mathcal{R}(i,j)$ corresponds to the background. To ensure labels corresponding to moving objects are not stored, label $\mathbf{b}_f(i,j)$ will be registered as $\mathbf{r}_{k+1}(i,j)$ only if it appears in at least f_{min} consecutive frames, where f_{min} ranges from 2 to 5.

In stage 2, representative sets $\mathcal{R}(i,j)$ having just one label are used to initialise the corresponding node locations $\mathcal{B}(i,j)$ in the background $\mathcal{B}$.

In stage 3, the remainder of the background is estimated iteratively. An optimal labelling solution is calculated by considering the likelihood of each of its labels along with the *a priori* knowledge of the local spatial neighbourhood modelled as an MRF. Iterated conditional mode (ICM), a deterministic relaxation technique, performs the optimisation.

The MRF framework is described in Section 3.2. The strategy for selecting the location of an empty background node to initialise a label is described in Section 3.3. The procedure for calculating the energy potentials, a prerequisite in

[1] For implementation purposes, each block location and its instances at every frame are treated as a node and its labels, respectively.

determining the *a priori* probability, is described in Section 3.4. In Section 3.5, the background model (used by the foreground segmentation algorithm overviewed in Section 2) is modified using the estimated background frame.

3.1 Similarity Criteria for Labels

Two labels $\mathbf{l}_f(i,j)$ and $\mathbf{r}_k(i,j)$ are similar if the following two constraints are satisfied:

$$\left\{(\mathbf{r}_k(i,j) - \mu_{r_k}(i,j))' \left(\mathbf{l}_f(i,j) - \mu_{b_f}(i,j)\right)\right\} \ / \ \{\sigma_{r_k}\sigma_{b_t}\} \ > \ \mathcal{T}_1 \tag{1}$$

and

$$\frac{1}{\phi N^2} \sum_{n=0}^{\phi N^2 - 1} |d_{k_n}(i,j)| \ < \ \mathcal{T}_2 \tag{2}$$

where μ_{r_k}, μ_{l_f} and $\sigma_{r_k}, \sigma_{l_f}$ are the mean and standard deviation of the elements of labels $\mathbf{r}_k$ and $\mathbf{l}_f$ respectively, while $\mathbf{d}_k(i,j) = \mathbf{l}_f(i,j) - \mathbf{r}_k(i,j)$.

Eqns. (1) and (2) respectively evaluate the correlation coefficient and the mean of absolute differences (MAD) between the two labels. The former constraint ensures that labels have similar texture/pattern while the latter one ensures that they are close in ϕN^2 dimensional space. In contrast, we note that in [6] the similarity criteria is based just on the sum of squared distances between the two blocks.

$\mathcal{T}_1$ is selected empirically (typically 0.8), to ensure that two visually identical labels are not treated as being different due to image noise. $\mathcal{T}_2$ is proportional to image noise.

3.2 Markov Random Field (MRF) Framework

MRF has been widely employed in solving problems in image processing that can be formulated as labelling problems [11,12].

Let $\mathbf{X}$ be a 2D random field, where each random variate $X_{(i,j)}$ $(\forall\, i,j)$ takes values in discrete *state space* Λ. Let $\omega \in \Omega$ be a *configuration* of the variates in $\mathbf{X}$, and let Ω be the set of all such configurations. The joint probability distribution of $\mathbf{X}$ is considered Markov if

$$p(\mathbf{X} = \omega) > 0, \ \forall\, \omega \in \Omega \tag{3}$$

and

$$p\left(X_{(i,j)} | X_{(a,b)}, (i,\, j) \neq (a,\, b)\right) = p\left(X_{(i,j)} | X_{\mathcal{N}_{(i,j)}}\right) \tag{4}$$

where $X_{\mathcal{N}_{(i,j)}}$ refers to the local *neighbourhood system* of $X_{(i,j)}$.

Unfortunately, the theoretical factorisation of the joint probability distribution of the MRF turns out to be intractable. To simplify and provide computationally efficient factorisation, Hammersley-Clifford theorem [13] states that an MRF can equivalently be characterised by a Gibbs distribution. Thus

$$p(\mathbf{X} = \omega) = e^{-U(\omega)/T} \ / \ \left(\sum\nolimits_{\omega} e^{-U(\omega)/T}\right) \tag{5}$$

where the denominator is a normalisation constant known as the *partition function*, T is a constant used to moderate the peaks of the distribution and $U(\omega)$ is an *energy function* which is the sum of *clique/energy potentials* V_c over all possible cliques C:

$$U(\omega) = \sum\nolimits_{c \in C} V_c(\omega) \tag{6}$$

The value of $V_c(\omega)$ depends on the local configuration of clique c.

In our framework, information from two disparate sources is combined using Bayes' rule. The local visual observations at each node to be labelled yield label likelihoods. The resulting label likelihoods are combined with *a priori* spatial knowledge of the neighbourhood represented as an MRF.

Let each input image I_f be treated as a realisation of the random field $\mathcal{B}$. For each node $\mathcal{B}(i,j)$, the representative set $\mathcal{R}(i,j)$ containing unique labels is treated as its *state space* with each $\mathbf{r}_k(i,j)$ as its plausible label[2].

Using Bayes' rule, the posterior probability for every label at each node is derived from the *a priori* probabilities and the observation-dependent likelihoods given by:

$$P(\mathbf{r}_k) = l(\mathbf{r}_k)p(\mathbf{r}_k) \tag{7}$$

The product is comprised of likelihood $l(\mathbf{r}_k)$ of each label $\mathbf{r}_k$ of set $\mathcal{R}$ and its *a priori* probability density $p(\mathbf{r}_k)$, conditioned on its local neighbourhood. In the derivation of likelihood function it is assumed that at each node the observation components $\mathbf{r}_k$ are conditionally independent and have the same known conditional density function dependent only on that node. At a given node, the label that yields maximum *a posteriori* (MAP) probability is chosen as the best continuation of the background at that node.

To optimise the MRF-based function defined in Eqn. (7), ICM is used since it is computationally efficient and avoids large scale effects[3] [11]. ICM maximises local conditional probabilities iteratively until convergence is achieved. In ICM an initial estimate of the labels is typically obtained by maximising the likelihood function. However, in our framework an initial estimate consists of partial reconstruction of the background at nodes having just one label which is assumed to be the background. Using the available background information, the remaining unknown background is estimated progressively (see Section 3.3).

At every node, the likelihood of each of its labels $\mathbf{r}_k$ $(k = 1, 2, \cdots, S)$ is calculated using corresponding weights W_k. The higher the occurrences of a label, the more is its likelihood to be part of the background. Empirically, the likelihood function is modelled by a simple weighted function, given by:

$$l(\mathbf{r}_k) = W_{c_k} / \sum\nolimits_{k=1}^{S} W_{c_k} \tag{8}$$

where $W_{c_k} = \min(W_{max}, W_k)$. Capping the weight is necessary in circumstances where the image sequence has a stationary foreground object visible for an exceedingly long period.

[2] To simplify the notations, index term (i,j) has been omitted from here onwards.

[3] An undesired characteristic where a single label is wrongly assigned to most of the nodes of the random field.

The spatial neighbourhood modelled as Gibbs distribution (Eqn. (5)) is encoded into an *a priori* probability density. The formulation of the clique potential $V_c(\omega)$ referred in Eqn. (6) is described in the Section 3.4. Using Eqns. (5) and (6) the calculated clique potentials $V_c(\omega)$ are transformed into *a priori* probabilities. For a given label, the smaller the value of energy function, the greater is its probability in being the best match with respect to its neighbours.

In our evaluation of the posterior probability given by Eqn. (7), more emphasis is given to the local spatial context term than the likelihood function which is based on mere temporal statistics. Thus, taking log of Eqn. (7) and assigning a weight to the prior, we get:

$$\log\left(P(\mathbf{r}_k)\right) = \log\left(l(\mathbf{r}_k)\right) + \eta \log\left(p(\mathbf{r}_k)\right) \tag{9}$$

where η has been empirically set to number of neighbouring nodes used in clique potential calculation (typically $\eta = 3$).

3.3 Node Initialisation

Nodes containing a single label in their representative set are directly initialised with that label in the background (see Fig. 1(ii)). However, in rare situations there's a possibility that all sets may contain more than 1 label (no trivial nodes). In such cases, the label having the largest weight from the representative sets of the 4 corner nodes is selected as an initial seed. We assume at least 1 of the corner regions corresponds to a static region. The rest of the nodes are initialised based on constraints as explained below. In our framework, the local *neighbourhood system* [14] of a node and the corresponding cliques are defined as shown in Fig. 2. The background at an empty node will be assigned only if at least 2 neighbouring nodes of its 4-connected neighbours adjacent to each other and the diagonal node located between them are already assigned with background labels. For instance, in Fig. 2, we can assign a label to node X if at least nodes B, D (adjacent 4-connected neighbours) and A (diagonal node) have already been assigned with labels. In other words, label assignment at node X is *conditionally independent* of all other nodes given these 3 neighbouring nodes.

Let us assume that all nodes except X are labelled. To label node X the procedure is as follows. In Fig. 2, four cliques involving X exist. For each candidate label at node X, the energy potential for each of the four cliques is evaluated independently given by Eqn. (10) and summed together to obtain its energy value. The label that yields the least value is likely to be assigned as the background.

Mandating that the background should be available in at least 3 neighbouring nodes located in three different directions with respect to node X ensures that the best match is obtained after evaluating the continuity of the pixels in all possible orientations.

In cases where not all the three neighbours are available, to assign a label at node X we use one of its 4-connected neighbours whose node has already been assigned with a label. Under these contexts, the clique is defined as two adjacent nodes either in the horizontal or vertical direction.

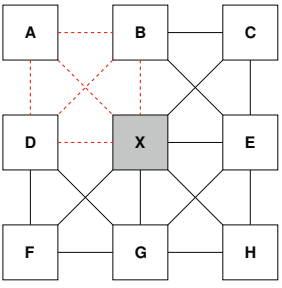


Fig. 2. The local neighbourhood system and its four cliques. Each clique is comprised of 4 nodes (blocks). To demonstrate one of the cliques, the the top-left clique has dashed red links.

After initialising all the empty nodes an accurate estimate of the background is typically obtained. Nonetheless, in certain circumstances an incorrect label assignment at a node may cause an error to occur and propagate to its neighbourhood. The problem is successfully redressed by the application of ICM. In subsequent iterations, in order to avoid redundant calculations, the label process is carried out only at nodes where a change in the label of one of their 8-connected neighbours occurred in the previous iteration.

3.4 Calculation of the Energy Potential

In Fig. 2, it is assumed that all nodes except X are assigned with the background labels. The algorithm needs to assign an optimal label at node X. Let node X have S labels in its state space $\mathcal{R}$ for $k = 1, 2, \cdots, S$, where one of them represents the true background. Choosing the best label is accomplished by analysing the spectral response of every possible clique constituting the unknown node X. For the decomposition we chose the Discrete Cosine Transform (DCT) [10] in a similar manner to [15].

We consider the top left clique consisting of nodes A, B, D and X. Nodes A, B and C are assigned with background labels. Node X is assigned with one of S candidate labels. For each colour channel z, we take the 2D DCT of the resulting clique. The transform coefficients are stored in matrix $\mathbf{T}_k^z$ of size $M \times M$ $(M = 2N)$ with its elements referred to as $T_k^z(v, u)$. The term $T_k^z(0, 0)$ (reflecting the sum of pixels at each node) is forced to 0 since we are interested in analysing the spatial variations of pixel values.

Similarly, for other labels present in the state space of node X, we compute their corresponding 2D DCT as mentioned above. A graphical example of the procedure is shown in Fig. 3.

Assuming that pixels close together have similar intensities, when the correct label is placed at node X, the resulting transformation has a smooth response (less high frequency components) when compared to other candidate labels.

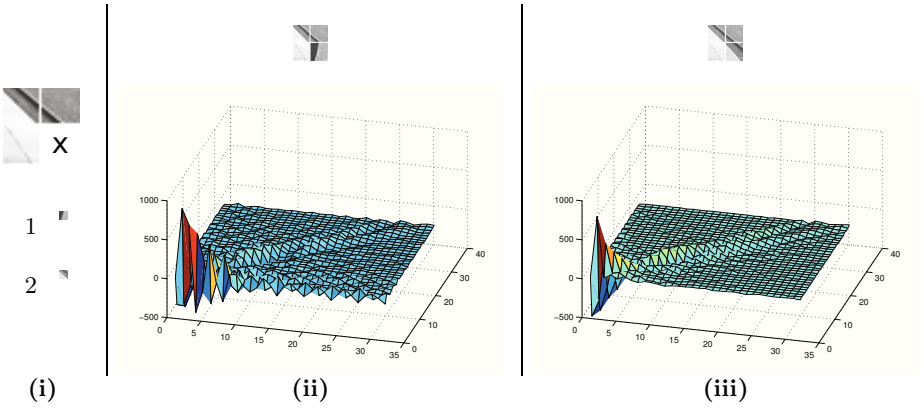


Fig. 3. An example of the processing done in Section 3.4. **(i)** A clique involving empty node X with two candidate labels in its representative set. **(ii)** A clique and a graphical representation of its DCT coefficient matrix where node X is initialised with candidate label 1. The gaps between the blocks are for ease of interpretation only and are not present during DCT calculation. **(iii)** As per (ii), but using candidate label 2. The smoother spectral distribution for candidate 2 suggests that it is a better fit than candidate 1.

The energy potential for each label is calculated after summing potentials obtained across the ϕ colour channels, as given below:

$$V_c(\omega_k) = \sum_{z=1}^{\phi} \left(\sum_{v=1}^{M} \sum_{u=1}^{M} |T_k^z(v,u)| \right) \tag{10}$$

where ω_k is the local configuration involving label k. The potentials over the other three cliques in Fig. 2 are calculated in a similar manner.

3.5 Modified Background Model for Foreground Segmentation

The foreground detection framework described in Section 2 uses a background model comprised of location specific multivariate Gaussians. The background image reconstructed through the MRF-based process is used as follows. First, the dual-Gaussian training strategy used in Section 2 is run on a given training sequence, obtaining the mean vectors and diagonal covariance matrices for each location. The mean vectors are then replaced by rerunning step 1 of the segmentation framework on the estimated background image. The covariance matrices are retained as is. Preliminary experiments indicated that when stationary backgrounds were occluded by foreground objects for a long duration, the variances computed in step 1 were similar to the variances of the true background.

Fig. 4. **(i)** Example frames from CAVIAR dataset; foreground masks obtained using: **(ii)** GMM based method [8], **(iii)** histogram based method [9], **(iv)** baseline multi-stage classifier [1], **(v)** proposed MRF based framework. We note the masks shown in columns **(ii)** to **(iv)** have considerable amount of false negatives since the foreground objects were included in the background model, while the results of the proposed framework (column **(v)**) have minimal errors.

4 Experiments

The proposed framework (MRF + multi-stage classifier) was evaluated with segmentation methods based on the baseline multi-stage classifier [1], Gaussian mixture models (GMMs) [8] and feature histograms [9]. In our experiments the same parameter settings were used across all sequences (i.e., they were not optimised for any particular sequence). The block size was set to 16×16. The values of $\mathcal{T}_1$ and $\mathcal{T}_2$ (see Eqns. 1 and 2) were set to 0.8 and 3 respectively, while W_{max} (see Eqn. 8) and T (Eqn. 5) were set to 150 and 1024 respectively. The algorithm was implemented in C++ with the aid of the Armadillo library [16].

We used the OpenCV v2.0 [17] implementations for the last two algorithms, in conjunction with morphological post-processing (opening followed by closing using a 3×3 kernel) in order to improve the quality of the obtained foreground masks [9]. The methods' default parameters were found to be optimal, except for the histogram method, where the built-in morphology operation was disabled as we found that it produced worse results than the above-mentioned opening and closing. We note that the proposed foreground segmentation approach does not require any such ad hoc post-processing.

In our experiments, we studied the influence of the various foreground segmentation algorithms on tracking performance. The foreground masks obtained from the detectors were passed as input to several tracking systems. We used the tracking systems implemented in the video surveillance module of OpenCV v2.0 [17] and the tracking ground truth data that is available for the sequences

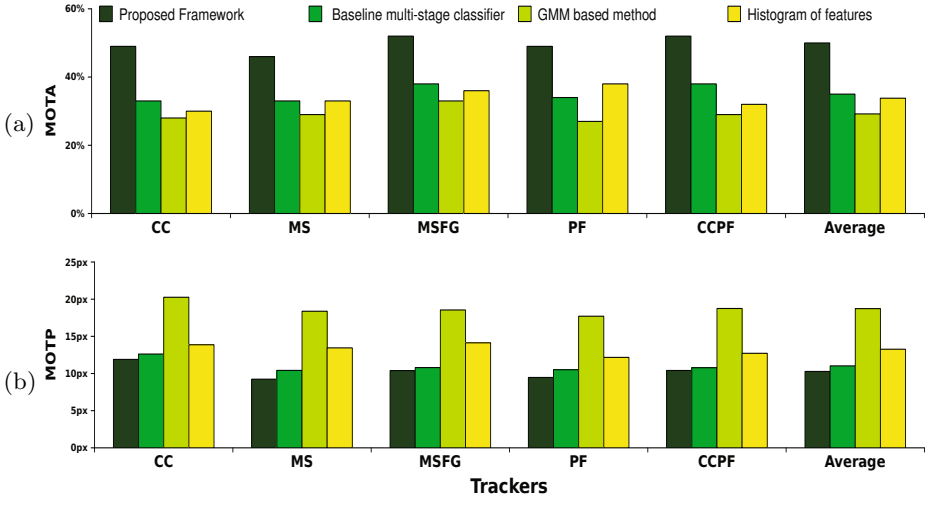


Fig. 5. Effect of foreground detection methods on: **(a)** multiple object tracking accuracy (MOTA), where taller bars indicate better accuracy; **(b)** multiple object tracking precision (MOTP), where shorter bars indicate better precision (lower distance). Results are grouped by tracking algorithm: blob matching (CC), mean shift trackers (MS and MSFG), particle filter (PF) and hybrid tracking (CCPF).

in the second set of the CAVIAR[4] dataset. We randomly picked 30 sequences from the dataset for our experiments. The tracking performance was measured with two metrics: multiple object tracking accuracy (MOTA) and multiple object tracking precision (MOTP), as proposed by Bernardin and Stiefelhagen [18].

Briefly, MOTP measures the average pixel distance between the ground-truth locations of objects and their locations according to a tracking algorithm. The lower the MOTP, the better. MOTA accounts for object configuration errors, false positives, misses as well as mismatches. The higher the MOTA, the better.

We performed 20 tracking simulations by evaluating four foreground object segmentation algorithms (baseline multi-stage classifier, GMM, feature histogram and the proposed method) in combination with five tracking algorithms (blob matching, mean shift, mean shift with foreground feedback, particle filter, and blob matching with particle filter for occlusion handling). The performance result in each simulation is the average performance of the 30 test sequences. We used the first 200 frames of each sequence for initialising the background model.

Examples of qualitative results are illustrated in Fig. 4. It can be observed that foreground masks generated using methods based on GMMs [8], feature histograms [9], and the baseline multi-stage classifier [1] have considerable false negatives, which are due to foreground objects being included into the background model. In contrast, the MRF based model initialisation approach results in noticeably better foreground detection.

[4] http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/

The quantitative tracking results, presented in Fig. 5, indicate that in all cases the proposed framework led to the best precision and accuracy values. For tracking precision (MOTP), the next best method [1] obtained an average pixel distance of 11.03, while the proposed method reduced the distance to 10.28, indicating an improvement of approximately 7%. For tracking accuracy (MOTA), the next best method obtained an average accuracy value of 0.35, while the proposed method achieved 0.5, representing a considerable improvement of about 43%.

5 Main Findings

In this paper we have proposed a foreground segmentation framework which effectively segments foreground objects in cluttered environments. The MRF-based model initialisation strategy allows the training sequence to contain foreground objects. We have shown that good background model initialisation results in considerably improved foreground detection, which leads to better tracking.

We noticed (via subjective observations) that all evaluated algorithms perform reasonably well when foreground objects are always in motion (i.e., where the background is visible for a longer duration when compared to the foreground). However, accurate estimation by methods solely relying on temporal statistics to initialise their background model becomes problematic if the above condition is not satisfied. This is the main area where the proposed framework is able to detect foreground objects accurately.

A minor limitation exists, as there is a potential to mis-estimate the background in cases where an occluding foreground object is smooth (uniform intensity value), has intensity value similar to that of the background (i.e., low contrast between the foreground and the background) and the true background is characterised by strong edges. Under these conditions, the energy potential of the label containing the foreground object is smaller (i.e., smoother spectral response) than that of the label corresponding to the true background. This limitation will be addressed in future work.

Overall, the parameter settings for the proposed algorithm appear to be quite robust against a variety of sequences and the method does not require explicit post-processing of the foreground masks. Experiments conducted to evaluate the effect on tracking performance (using the CAVIAR dataset) show the proposed framework obtains considerably better results (both qualitatively and quantitatively) than approaches based on Gaussian mixture models (GMMs) [8] and feature histograms [9].

Acknowledgements. NICTA is funded by the Australian Government as represented by the *Department of Broadband, Communications and the Digital Economy* as well as the Australian Research Council through the *ICT Centre of Excellence* program.

References

1. Reddy, V., Sanderson, C., Sanin, A., Lovell, B.: Adaptive Patch-Based Background Modelling for Improved Foreground Object Segmentation and Tracking. In: Proc. Advanced Video and Signal Based Surveillance (AVSS), pp. 172–179 (2010)
2. Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: Principles and practice of background maintenance. In: Proc. Int. Conf. Computer Vision (ICCV), vol. 1, pp. 255–261 (1999)
3. Kim, K., Chalidabhongse, T., Harwood, D., Davis, L.: Real-time foreground–background segmentation using codebook model. Real-Time Imaging 11, 172–185 (2005)
4. Maddalena, L., Petrosino, A.: A Self-Organizing Approach to Background Subtraction for Visual Surveillance Applications. IEEE Trans. Image Processing 17, 1168–1177 (2008)
5. Wang, H., Suter, D.: A Novel Robust Statistical Method for Background Initialization and Visual Surveillance. In: Narayanan, P.J., Nayar, S.K., Shum, H.-Y. (eds.) ACCV 2006. LNCS, vol. 3851, pp. 328–337. Springer, Heidelberg (2006)
6. Colombari, A., Fusiello, A., Murino, V.: Background Initialization in Cluttered Sequences. In: CVPRW, Washington DC, USA, pp. 197–202 (2006)
7. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. In: Proc. Intl. Conf. Computer Vision (ICCV), vol. 1, pp. 377–384 (1999)
8. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: Proc. Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 246–252 (1999)
9. Li, L., Huang, W., Gu, I., Tian, Q.: Foreground object detection from videos containing complex background. In: ACM Int. Conf. Multimedia, pp. 2–10 (2003)
10. Gonzales, R., Woods, R.: Digital Image Processing, 3rd edn. Prentice-Hall, Englewood Cliffs (2007)
11. Besag, J.: On the statistical analysis of dirty images. Journal of Royal Statistics Society 48, 259–302 (1986)
12. Sheikh, Y., Shah, M.: Bayesian Modeling of Dynamic Scenes for Object Detection. IEEE Trans. Pattern Analysis and Machine Intelligence 27, 1778–1792 (2005)
13. Besag, J.: Spatial interaction and the statistical analysis of lattice systems. Journal of the Royal Statistical Society. Series B (Methodological) 32, 192–236 (1974)
14. Geman, S., Geman, D.: Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. IEEE Trans. Pattern Analysis and Machine Intelligence, 721–741 (1984)
15. Reddy, V., Sanderson, C., Lovell, B.: An efficient and robust sequential algorithm for background estimation in video surveillance. In: Proc. Int. Conf. Image Processing (ICIP), pp. 1109–1112 (2009)
16. Sanderson, C.: Armadillo: An open source C++ linear algebra library for fast prototyping and computationally intensive experiments. Technical report, NICTA (2010)
17. Bradski, G., Kaehler, A.: Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media, Sebastopol (2008)
18. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: the CLEAR MOT metrics. EURASIP Journal on Image Video Processing (2008)

Adaptive εLBP for Background Subtraction

LingFeng Wang[1], HuaiYu Wu[2], and ChunHong Pan[1]

[1] NLPR, Institute of Automation, Chinese Academy of Sciences
lfWang@nlpr.ia.ac.cn, chpan@nlpr.ia.ac.cn
[2] Peking University, Key Laboratory of Machine Perception (MOE)
huaiyuwu@gmail.com

Abstract. Background subtraction plays an important role in many computer vision systems, yet in complex scenes it is still a challenging task, especially in case of illumination variations. In this work, we develop an efficient texture-based method to tackle this problem. First, we propose a novel adaptive ε**LBP** operator, in which the threshold is adaptively calculated by compromising two criterions, i.e. the description stability and the discriminative ability. Then, the naive Bayesian technique is adopted to effectively model the probability distribution of local patterns in the pixel level, which utilizes only one single ε**LBP** pattern instead of ε**LBP** histogram of local region. Our approach is evaluated on several video sequences against the traditional methods. Experiments show that our method is suitable for various scenes, especially can robust handle illumination variations.

1 Introduction

Background subtraction is an effective method to detect foreground objects from a stationary camera. The central idea behind this method is to utilize the visual properties of the scene to build an appropriate representation, which is then be used to classify a new observation as foreground or background. Existing methods for background modeling can be mainly classified into two categories, i.e. the *predictive* methods and the *statistical* methods. In the *predictive* methods, the scene is modeled as a time series and a dynamical model is developed to recover the current state based on past observations [1,2,3,4]. In the *statistical* methods, the order of the input observations is neglected, and a probabilistic representation of the observations is then roughly built [5,6,7,8,9,10,11]. The background subtraction method proposed in this paper belongs to the latter.

A popular *statistical* method is to model each background pixel with a single Gaussian distribution [5]. However, This method does not work well in the case of dynamic natural environments with repetitive motions, i.e. waving vegetation, rippling water, and camera jitter. In [6], the mixture of Gaussians(**GMM**) approach is proposed to handle these complex, non-static backgrounds. Unfortunately, background with fast variations can not be accurately modeled by just a few Gaussians. To overcome the limitations of parametric methods (i.e. single Gaussian in [5], **GMM** in [6]), a nonparametric technique is developed

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 560–571, 2011.

in [7,12]. [7] utilizes a kernel density estimation technique to build a statistical representation of the scene background. [12] uses a codebook to construct a compressed background model. However, whether the parametric methods or the nonparametric methods may fail when foreground objects have similar color as background, or when the illumination conditions vary with sunlight changing outdoor or light switching indoor. The main reason is that these methods only use the pixel color or intensity information to detect foreground objects. To deal with this weak description, [8,9,10] use a novel and powerful approach based on discriminative texture features represented by **LBP** [13] to capture background statistics. In [8], each pixel is modeled as a group of **LBP** histograms that are calculated over a circular region around the pixel. The main limitation of this method is that both memories and computation costs significantly increase with the increasing of the images resolution. Moreover, the descriptive ability of **LBP** is not very robust, since the **LBP** is sensitive to the noise. To overcome these limitations, ε**LBP** [10] (scale invariance **LBP**) is proposed, in which a small threshold is added into the traditional **LBP** description. Unfortunately, this threshold is empirically selected as a global constant. Thus, this method only perform well when the illumination variation is global.

In this work, we propose an efficient background subtraction framework that addresses local illumination variations on the feature level. We improve the ε**LBP** by adding local adaptive property. The threshold ε is adaptively selected for each pair of two neighboring pixels. With two evaluation criterions proposed, a simple yet effective approach is presented to adaptively estimate the threshold by classifying all the pixels into two groups, i.e. the edge pixels and the texture pixels. In background modeling procedure, the naive Bayesian technique is adopted to effectively model the probability distribution of local patterns in the pixel level. The utilization of single ε**LBP** (pixel level) improves the robustness to the illumination variation and reduces the computation cost compared with the state-of-the-art, such as the **GMM** and the Codebook.

The rest of this paper is organized as follows. A brief introduction about adaptive ε**LBP** is given in Section 2. The background subtraction method is described in Section 3 in detail. Experiment results on real data compared with the traditional methods are reported in Section 4. The conclusive remark is addressed at the end of this paper.

2 From LBP to Adaptive εLBP

LBP [14,13] is a texture primitive statistic, which is gray-scale invariant. This operator labels the pixels of an image by thresholding the value of center pixel with its neighborhoods and encoding the result as a binary number:

$$\boldsymbol{\pi} = \mathbf{LBP}_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c)2^p, \qquad s(t) = \begin{cases} 1 & t \geqslant 0 \\ 0 & t < 0 \end{cases} \tag{1}$$

where g_c corresponds to the gray value of the center pixel (x_c, y_c), and g_p corresponds to the gray values of P equally spaced pixels on a circle of radius R.

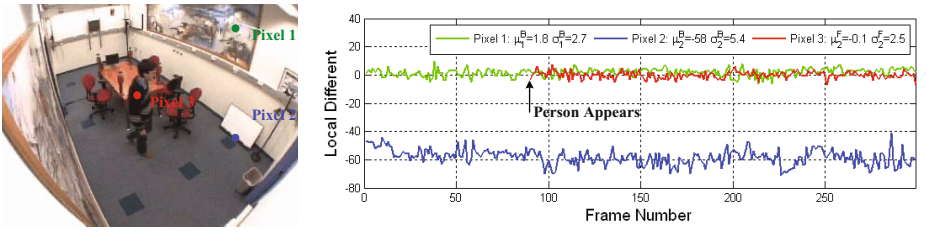


Fig. 1. Overview of three local difference sequences along with time. **Pixel 1** and **Pixel 2** are in background (in texture area and edge respectively), and **Pixel 3** is in foreground object (fixed in person's shoulder as person walking).

The advantage of the **LBP** operator is the tolerance against the illumination changes and the computational simplicity. However, the **LBP** operator is not robust to local image noises when neighbor pixels are similar. The ε**LBP** [10] are proposed by adding a small scaling threshold ε to handle the local noises. As mentioned in [10], using the threshold instead of a small offset can solve the intensity invariant problem, which is caused by illumination variance. Unfortunately, the threshold ε is empirically selected as a global constant. Note that, the adaptive property of the threshold is very important, because the illumination variations are often locally. Thus, it is irrational to use a global constant as the threshold. In this work, we first propose two criterions to evaluate the threshold ε, and then propose a simple but effective method to estimate ε adaptively.

Actually, a binary pattern is obtained by modeling the local difference between two neighboring pixels. As shown in Fig.1, for a fixed pixel in background (**Pixel 1** or **Pixel 2**), the local difference forms a distribution (background distribution). Similarly, for a fixed pixel in foreground object (**Pixel 3**), such as in person's shoulder in Fig.1, the local difference also forms a distribution (foreground distribution). Here we assume these two distributions satisfy the single Gaussian model, and denote them as $\mathcal{N}(\mu^{\mathrm{B}}, \sigma^{\mathrm{B}})$ and $\mathcal{N}(\mu^{\mathrm{F}}, \sigma^{\mathrm{F}})$.

Left sub-figure of Fig.2 gives an example of binary patterns formed by different thresholds. Red and blue curves represent the foreground distribution and the background distribution, respectively, and three vertical lines denote three thresholds (in different colors). For example, we use the yellow vertical line to represent the threshold. If the pixel value falls on the left side of the yellow vertical line, the binary pattern is formed as 0. Otherwise, the binary pattern is formed as 1. In this figure, the pixel value falls on the left side means that this pixel value is smaller than **-10**. The main difference among the **LBP**, ε**LBP**, and adaptive ε**LBP** methods is the way to calculate a rational threshold. For example, the original **LBP** approach simply adopts 0 as the threshold. Thus, the original **LBP** selects the green vertical line as the threshold. To describe our method, we present two binary patterns definitions, i.e. the background binary pattern and the foreground binary pattern. If the current pixel lies in background (e.g. **Pixel 1** or **Pixel 2** in Fig.1), the formed binary pattern is denoted as the background binary pattern. Otherwise, if the current pixel lies in

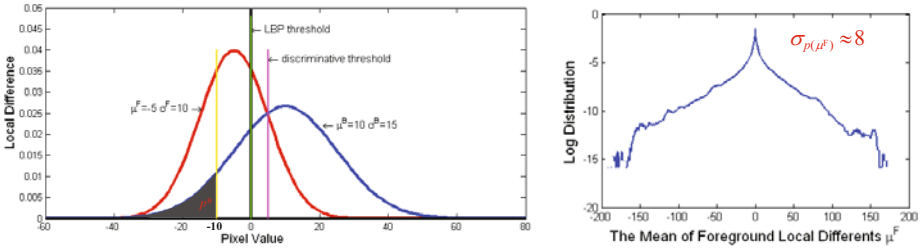


Fig. 2. Left sub-figure gives an example of forming binary patterns by different thresholds. Right sub-figure shown the log distribution of μ^{F}.

foreground object (e.g. **Pixel 3** in Fig.1), the formed binary pattern is denoted as the foreground binary pattern. Then, we use two criterions to determine the threshold. First, the description stability for both the background binary pattern and the foreground binary pattern. Second, the discriminative ability between the background binary pattern and the foreground binary pattern.

When the threshold and the background distribution are known, we can obtain the probability that the background binary pattern is 0. We denote this probability as p^{B}. As illustrated Fig.2, if the threshold lies on the yellow vertical line, the shadow region represents this probability. The description stability is fully determined by the probability p^{B}. For example, if the $p^{\mathrm{B}} = 1$, the background binary pattern can only be formed as 0. That is, the description is absolutely stable. If the $p^{\mathrm{B}} = 0.5$, the background binary pattern will be formed as 0 or 1 randomly. That is, the description is very unstable. We use entropy to define stability of the background binary pattern, given by

$$\mathcal{H}^{\mathrm{B}} = \mathcal{H}(p^{\mathrm{B}}|\mu^{\mathrm{B}}, \sigma^{\mathrm{B}}, \varepsilon) = -p^{\mathrm{B}} \log p^{\mathrm{B}} - (1 - p^{\mathrm{B}}) \log (1 - p^{\mathrm{B}}) \tag{2}$$

From Eqn.2, we see that if the probability p^{B} become small, the entropy $\mathcal{H}^{\mathrm{B}}$ will become small too. That is, when the entropy is smaller, the description is more stable. Same as the definition of $\mathcal{H}(p^{\mathrm{B}}|\mu^{\mathrm{B}}, \sigma^{\mathrm{B}}, \varepsilon)$, for a specified foreground distribution, the description stability in the foreground binary pattern can be also defined as $\mathcal{H}(p^{\mathrm{F}}|\mu^{\mathrm{F}}, \sigma^{\mathrm{F}}, \varepsilon)$. However, for a fixed pixel, we do not know which foreground pixel will appear beforehand. That is to say, the foreground distribution is unknown in advance. Here, we assume the parameters $\mu^{\mathrm{F}}, \sigma^{\mathrm{F}}$ follow a distribution $p(\mu^{\mathrm{F}}, \sigma^{\mathrm{F}})$. Therefore, description stability in foreground can be defined as the expectation of the entropy:

$$\mathcal{H}^{\mathrm{F}} = \mathbb{E}_{p(\mu^{\mathrm{F}}, \sigma^{\mathrm{F}})} \Big(\mathcal{H}(p^{\mathrm{F}}|\mu^{\mathrm{F}}, \sigma^{\mathrm{F}}, \varepsilon) \Big) = \iint \mathcal{H}(p^{\mathrm{F}}|\mu^{\mathrm{F}}, \sigma^{\mathrm{F}}, \varepsilon) p(\mu^{\mathrm{F}}, \sigma^{\mathrm{F}}) \mathrm{d}\mu^{\mathrm{F}} \mathrm{d}\sigma^{\mathrm{F}} \tag{3}$$

For a specified foreground distribution, the discrimination between background binary pattern and foreground binary pattern can be defined as the Bayesian error $\mathcal{E}(p^{\mathrm{F}}, p^{\mathrm{B}}) = p^{\mathrm{B}} + 1 - p^{\mathrm{F}}$. Therefore, the whole discrimination is defined as the expectation of the Bayesian error:

$$\mathcal{E} = \mathbb{E}_{p(\mu^{\mathrm{F}},\sigma^{\mathrm{F}})}\Big(\mathcal{E}(p^{\mathrm{F}},p^{\mathrm{B}})\Big) = \iint \mathcal{E}(p^{\mathrm{F}},p^{\mathrm{B}})p(\mu^{\mathrm{F}},\sigma^{\mathrm{F}})\mathrm{d}\mu^{\mathrm{F}}\mathrm{d}\sigma^{\mathrm{F}} \tag{4}$$

For a specified foreground distribution, the pink vertical line is the minimum Bayesian error threshold (the discriminative threshold). Combining the Eqn.2, Eqn.3, and Eqn.4, the optimal ε can be obtained by minimizing the Eqn.5:

$$\varepsilon = \arg\min_{\varepsilon} \lambda_1\mathcal{H}^{\mathrm{B}} + \lambda_2\mathcal{H}^{\mathrm{F}} + \lambda_3\mathcal{E} \tag{5}$$

where $\lambda_{1,2,3}$ are three weights. It is almost infeasible to calculate the optimal ε from the Eqn.4, because it is difficult to get the exact density $p(\mu^{\mathrm{F}},\sigma^{\mathrm{F}})$ and to choose three appropriate weights λ_1, λ_2, and λ_3.

In this work, σ^{F} is simplified as a constant, thus the density $p(\mu^{\mathrm{F}},\sigma^{\mathrm{F}})$ is simplified as $p(\mu^{\mathrm{F}})$. We define $p(\mu^{\mathrm{F}})$ as the foreground mean distribution. As shown in Fig.2, we count the distribution $p(\mu^{\mathrm{F}})$ and compute its variance $\mathbb{D}\big(p(\mu^{\mathrm{F}})\big) = \sigma_{p(\mu^{\mathrm{F}})} \approx 8$. In order to calculate Eqn.5 simply, we first classify all pixels into two groups, i.e. the edge pixels (e.g. **Pixel 2** in Fig.1) and texture pixels (e.g. **Pixel 1** in Fig.1). For the edge pixel, the pixel value is extremely different from its neighborhoods. Otherwise, for the texture pixel, the pixel value is similar with its neighborhoods. Then, we calculate the thresholds of the two groups with different methods, as shown in Fig.3. For the edge pixels, the discriminative ability is considered more important than the stability. For the texture pixels, the stability is more important that the discriminative ability. Furthermore, the threshold ε should not be too small or too large. Accordingly, for the edge pixels, when $|\mu^{\mathrm{B}}| > \alpha \cdot \sigma_{p(\mu^{\mathrm{F}})}$, the threshold ε_c is calculated:

$$\varepsilon_c = \begin{cases} \max\left(\eta, \frac{\mu^{\mathrm{B}} - \gamma\cdot\sigma^{\mathrm{B}}}{g_c}\right) & \mu^{\mathrm{B}} > \alpha \cdot \sigma_{p(\mu^{\mathrm{F}})} \\ \min\left(-\eta, \frac{\mu^{\mathrm{B}} - \gamma\cdot\sigma^{\mathrm{B}}}{g_c}\right) & \mu^{\mathrm{B}} < -\alpha \cdot \sigma_{p(\mu^{\mathrm{F}})} \end{cases} \tag{6}$$

where α, γ, and η are the constants, g_c corresponds to the gray value of the center pixel, and the max(.) and min(.) operators are used to restrict the threshold. For the texture pixels, when $|\mu^{\mathrm{B}}| \leq \alpha \cdot \sigma_{p(\mu^{\mathrm{F}})}$, the threshold ε_c is calculated:

$$\varepsilon_c = \begin{cases} -\eta & \mu^{\mathrm{B}} \leq \alpha \cdot \sigma_{p(\mu^{\mathrm{F}})} \ \&\ \mu^{\mathrm{B}} \geq 0 \\ \eta & \mu^{\mathrm{B}} \geq -\alpha \cdot \sigma_{p(\mu^{\mathrm{F}})} \ \&\ \mu^{\mathrm{B}} < 0 \end{cases} \tag{7}$$

Since the threshold of each pixel with its neighborhoods are gained, the adaptive ε**LBP** of this pixel can be calculated as follow,

$$\boldsymbol{\pi} = \varepsilon\mathbf{LBP}_{P,R}(x_c,y_c) = \sum_{p=0}^{P-1} s\left(\frac{g_p - g_c}{g_c} - \varepsilon_c^p\right)2^p \tag{8}$$

where ε_c^p is the threshold of the p-th neighborhood of the center pixel.

From Fig.3, we can get some conclusions. First, η is used to prevent that the threshold is too small. Second, for the edge pixel, the threshold tends to lie between the background distribution peak and the foreground mean distribution peak. And for the texture pixels, the threshold tends to lie on same side

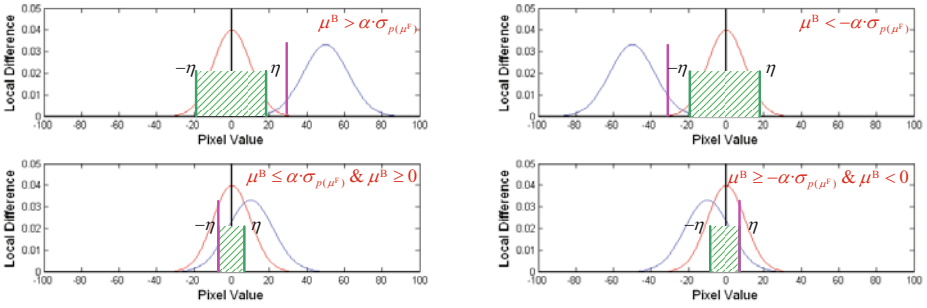


Fig. 3. Overview four cases of threshold selection. The up two sub-figures show the edge pixels, while the down two sub-figures show the texture pixels. For each sub-figure, the red equation on the upright represents the condition. The blue curve represents the background distributions. The red curve the foreground mean distribution $p(\mu^F)$. The green shadow region indicates the restricted region for the threshold calculation. That is, the absolute value of the threshold should not be smaller than η. The pink vertical line gives the result threshold. Note that, the restricted regions are different from each other, since the mean μ^B of each pixel is different.

of the two distribution peaks. From the above descriptions, we know that the foreground mean distribution reflects the overall nature of the foreground distribution. From the figure, the Bayesian error of the two distributions on the edge pixel is smaller than on the texture pixel. Thus, the threshold on edge pixel is more discriminative than on the texture pixel. As shown in the Eqn.6 and Eqn.7, we can see the threshold ε_c is only determined by the mean μ^B. When performing background subtraction, μ^B is first obtained from the start n frames. That is, we adopt the mean or median of these n frames as μ^B. Then, we use Eqn.6 and Eqn.7 to calculate the thresholds. After that, for each input frame, we use Eqn.8 to compute the adaptive ε**LBP** of each pixel.

3 Background Subtraction Based on Adaptive εLBP

The flow chart of our background subtraction method is summarized in Fig.4. When performing online background subtraction, an input image is first analyzed by calculating the adaptive ε**LBP** of each pixel based on Eqn.8. Then, the probability of each pixel belonging to background is calculated based on Eqn.12. Finally, the foreground mask is obtained by thresholding this probability, as shown in the Eqn.13. Meanwhile, the model is updated based on Eqn.14.

In order to segment the foreground, we consider to model the background in a pixel-wise manner. Given an input video sequence, the pixel process with the adaptive ε**LBP** observations over time $1, 2, \ldots, t$ at a location (x_c, y_c) is defined as $\{\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \ldots, \boldsymbol{\pi}_t\}$. Each pattern $\boldsymbol{\pi}_i$ is composed of the P neighbor binary patterns, i.e. $\boldsymbol{\pi}_i = \{\pi_i^1, \pi_i^2, \ldots, \pi_i^P\}$. We define the background model as $\mathcal{B}$, which represents the ε**LBP** distribution. Given a new ε**LBP** observation $\widehat{\boldsymbol{\pi}}$, the probability of belonging to the background model $\mathcal{B}$ is defined as $p(\mathcal{B}|\widehat{\boldsymbol{\pi}})$. By

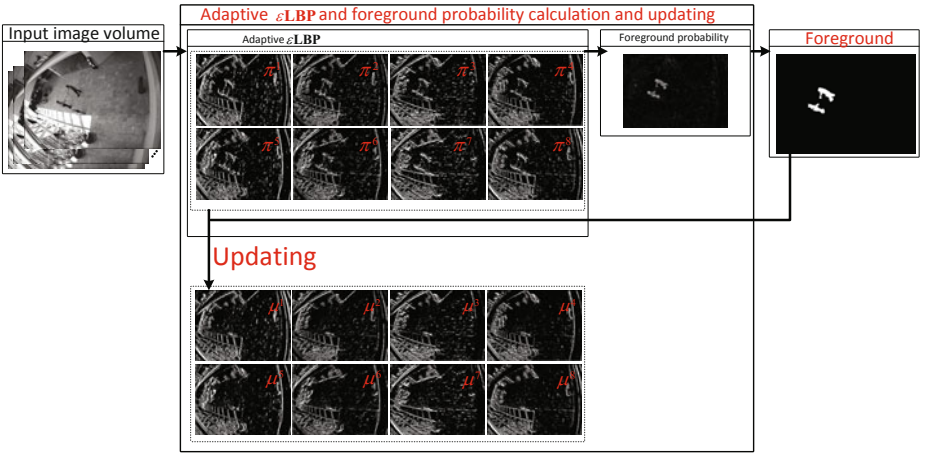


Fig. 4. Overview of our background subtraction algorithm. Each pixel in the up eight small images indicates a binary pattern π^i ($P = 8$ neighbors). And each pixel in the down eight images indicates a background model μ^i.

using the Bayesian rule and the naive conditional independence, which assumes each binary pattern $\widehat{\pi}^i$ is conditionally independent of every other pattern $\widehat{\pi}^j$ for $j \neq i$, we can get that

$$\begin{aligned} \varphi = p(\mathcal{B}|\widehat{\boldsymbol{\pi}}) &= \frac{p(\mathcal{B})p(\widehat{\boldsymbol{\pi}}|\mathcal{B})}{p(\boldsymbol{\pi})} = \frac{p(\mathcal{B})}{p(\boldsymbol{\pi})} p(\widehat{\pi}^1, \widehat{\pi}^2, \ldots, \widehat{\pi}^P|\mathcal{B}) \qquad \texttt{Bayesian} \\ &= \frac{p(\mathcal{B})}{p(\boldsymbol{\pi})} \prod_{i=1}^{P} p(\widehat{\pi}^i|\mathcal{B}) \qquad \texttt{Naive Conditional Independence} \end{aligned} \tag{9}$$

The prior $p(\mathcal{B})$ and the evidence $p(\boldsymbol{\pi})$ only affect the selection of the foreground threshold τ. The foreground threshold τ is used in Eqn.13, and it is usually chose empirically. Thereby, we only need to calculate $p(\widehat{\pi}^i|\mathcal{B})$. From the naive conditional independence assumption, each binary pattern π^i is conditionally independent others. Similarly, the i^{th} binary pattern's model can be also considered as conditionally independent others. Here, we use $\mathcal{B}^i$ to denote the i^{th} binary pattern's model. Thus, the conditional independence of binary pattern's model can be represented as follows: each binary pattern's model $\mathcal{B}^i$ is conditionally independent of every other pattern's model $\mathcal{B}^j$ for $j \neq i$. Background model $\mathcal{B}$ is the combination of all $\mathcal{B}^i$, given by $\mathcal{B} = \{\mathcal{B}^1, \mathcal{B}^2, \ldots, \mathcal{B}^P\}$. Thus, the probability $p(\widehat{\pi}^i|\mathcal{B})$ can be computed with

$$p(\widehat{\pi}^i|\mathcal{B}) = p(\widehat{\pi}^i|\mathcal{B}^1, \ldots, \mathcal{B}^i, \ldots, \mathcal{B}^P) = p(\widehat{\pi}^i|\mathcal{B}^i) \tag{10}$$

In this work, we adopt single Gaussian distribution to model $\mathcal{B}^i$. As is known, the binary pattern is the binomial distribution with the domain $\{0, 1\}$. Thus, we can only need to use the mean μ^i to model the single Gaussian distribution,

because the variance can be calculated by $\sigma^i = \sqrt{\mu^i(1-\mu^i)}$. For a new binary pattern observation $\widehat{\pi}^i$, the probability $p(\widehat{\pi}^i|\mathcal{B}^i)$ can be calculated as follow,

$$p(\widehat{\pi}^i|\mathcal{B}^i) = \frac{|\widehat{\pi}^i - \mu^i|}{\sigma^i} = \frac{|\widehat{\pi}^i - \mu^i|}{\sqrt{\mu^i(1-\mu^i)}} \tag{11}$$

Combining the Eqn.9, the Eqn.10, and the Eqn.11, we can get that,

$$\varphi = \frac{p(\mathcal{B})}{p(\boldsymbol{\pi})} \prod_{i=1}^{P} p(\widehat{\pi}^i|\mathcal{B}) \propto \prod_{i=1}^{P} p(\widehat{\pi}^i|\mathcal{B}) \propto \prod_{i=1}^{P} p(\widehat{\pi}^i|\mathcal{B}^i) \propto \prod_{i=1}^{P} \frac{|\widehat{\pi}^i - \mu^i|}{\sqrt{\mu^i(1-\mu^i)}} \tag{12}$$

Since the probability of each pixel belonging to background is calculated by Eqn.12, the foreground mask is gained by thresholding this probability,

$$m = \begin{cases} 1 & \varphi > \tau \\ 0 & \texttt{else} \end{cases} \tag{13}$$

where τ is a threshold. The updating procedure is required after obtaining foreground mask. We adopt the selective background updating approach, which is proposed in [15]. Specifically, μ^i is updated by the new observation $\widehat{\pi}^i$,

$$\mu^i = m\mu^i + (1-m)\left(\beta\mu^i + (1-\beta)\widehat{\pi}^i\right) \tag{14}$$

where β is a user-settable learning rate and m is the foreground mask.

4 Experimental Results

A number of data-sets are used to evaluate quantitatively and qualitatively the performance of the proposed scheme. We compare with several other approaches, i.e. the **GMM** [6], codebook [12], **LBP** [8], and ε**LBP** [10] methods. In the experiments, the neighborhood size is empirically selected by 3×3, that is $P = 8$ neighborhoods are used, and parameters are set as $\alpha = 5$, $\gamma = 2.5$, $\eta = 0.15$, $\tau = 2^P$, and $\beta = 0.1$ respectively. We use simple operations to further refine the output masks of all algorithms. For example, we use the erosion operation to remove the noise in the mask, and use the dilation operation to fill the small hole in the mask.

4.1 Qualitative Evaluation

The qualitative evaluation gives an insight of the results, which indicate the ways the foreground were segmented. Fig.5 presents the visual comparison of the proposed method with other methods. We also present the ground truth. The video sequences [16] are downloaded from the web site[1]. The video sequences include both indoor and outdoor scenes. From the results, it can be seen that the performances of the proposed method are better than the others. The quantitative

[1] http://research.microsoft.com/en-us/um/people/jckrumm/wallflower/testimages.htm

Fig. 5. The visual comparison of our method with others (including the ground truth). The images resolution are 160×120 pixels.

evaluation of these sequences are shown in Tab.1 in the Subsection 4.2 in detail. In the **Light Switch** sequence, the illumination condition is obviously changed when the person switches the light. Those background subtraction methods that rely only on color information, such as the **GMM** and codebook would fail to detect the moving object correctly. The main reason is that the colors of the foreground and the background are both changed significantly. Accordingly, it is hard to segment the foreground only use the color information. That is, texture information always plays an important role in background subtraction. Moreover, as shown in Fig.5, the **LBP** and ε**LBP** methods output some false negatives on the inner areas of the moving object. The definition of false negative is illustrated in the next subsection in detail. The main reason of this drawback is that the threshold is not well selected. By contrast, the proposed adaptive ε**LBP** method produces better results than others. The main reasons are from two aspects. First, it exploits texture information instead of a single pixel value. Second, it adopts the adaptive threshold, which optimally distinguish from the background to the foreground.

4.2 Quantitative Evaluation

The quantitative evaluation is done on the image sequences based on the F-score. The F-score measures the segmentation accuracy by considering both the recall and the precision, which is defined as

$$F = \frac{2 \cdot \mathrm{TP}}{2 \cdot \mathrm{TP} + \mathrm{FN} + \mathrm{FP}} \tag{15}$$

where TP, FP, and FN are the true positives (true foreground pixels), false positives (the number of background pixels marked as foreground pixels), and false negatives (the number of foreground pixels that are missed), respectively. Tab.1 gives the numerical comparison of the proposed method with others. Note that, the test sequences are same as those used in the Subsection.4.1. As shown in Tab.1, our results are better than the others. We also use another video sequence[2] to evaluate the proposed method. This video is challenging is challenging, because evident shadow follows the walking person. That is, the illumination condition is very unstable in this sequence. As shown in Fig.6, the results of **LBP**-like methods, i.e. the **LBP**, the ε**LBP**, and the adaptive ε**LBP**, are better than other two classical methods, i.e. the **GMM** and the codebook. Furthermore, the proposed adaptive ε**LBP** is better than the **LBP** and the ε**LBP**. To sum up, **LBP**-like methods are more robust to the illumination variation than the classical methods, and the adaptive ε**LBP** is better than other **LBP**-like methods.

Table 1. Overview of the comparison of all methods (F-score)

Algorithm	GMM	Codebook	**LBP**	ε**LBP**	Adaptive ε**LBP**
Bootstrap	0.452	0.665	0.611	0.766	0.800
Camouflage	0.979	0.986	0.897	0.983	0.982
Light Switch	0.280	0.318	0.582	0.518	0.773
Time of Day	0.918	0.909	0.760	0.905	0.935
Waving Trees	0.893	0.938	0.738	0.760	0.923

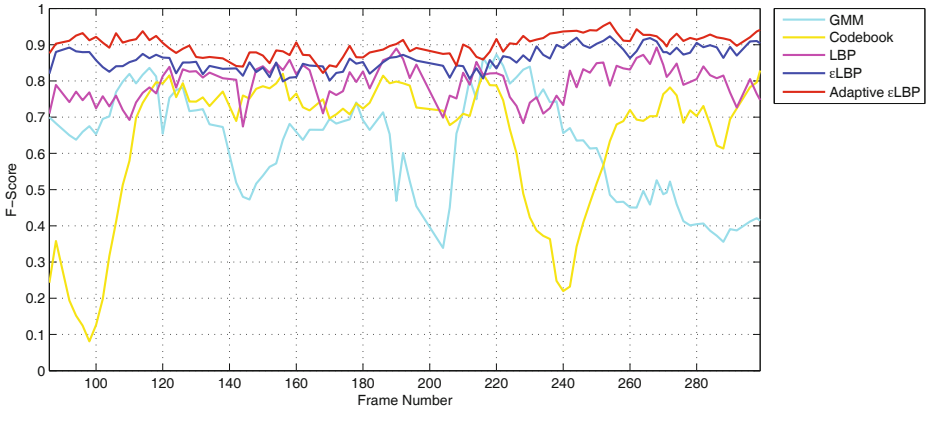


Fig. 6. Comparison results of our method with the traditional methods. The test image sequence exists the obvious shadow where person walking.

[2] http://cvrr.ucsd.edu/aton/shadow/data/intelligentroom.AVI

Table 2. Overview of the comparison of all methods (speed)

Algorithm	GMM	Codebook	**LBP**	ε**LBP**	Adaptive ε**LBP**
Speed	15.6(fps)	18.2	10.1	58.4	57.9

4.3 Computation Cost

The computation cost of all five methods is shown in Tab.2. The video sequence [17] is download from the web-site[3]. The resolution of these sequences are 320×240. The speed is measured by fps (frame per second). All algorithms are implemented by the MATLAB, and no special optimization is adopted. As shown in Tab.2, the speed of the proposed method is significant faster than other algorithms except the ε**LBP** method. Especially, the speed of the proposed method is nearly 6 times faster than the **LBP** method.

5 Conclusion

In this paper, we propose an efficient texture-based background subtraction method. The main contributions in this work can be summarized as follows. First, we propose an improved adaptive ε**LBP** operator, which is less sensitive to the illumination variations. The threshold ε is novelly calculated by compromising two proposed criterions. Second, we apply the adaptive ε**LBP** operator to background subtraction. The naive Bayesian technique is adopted to construct a pixel-based process model instead of **LBP** histogram. Our proposed background subtraction method possesses several advantages. For example, it overcome the sensitive to illumination variation in **GMM** and codebook methods. It improves the limitation of the computation cost in the **LBP** histogram method. Moreover, the threshold is calculated more accurately than the ε**LBP**. Note that, the disadvantage of our method is that the background image can not be gained, which is the common disadvantage of all the texture-based method. Fortunately, in many computer vision tasks only the foreground mask is required. In the future, we will add other features, such as the spectral, spatial, and temporal features [11], into the proposed background subtraction framework. Moreover, besides background subtraction, the improved adaptive ε**LBP** operator also can be applied on other fields, such as detection, tracking, and recognition.

Acknowledgement. This work was supported by the National Natural Science Foundation of China (Grant No. 60873161 and Grant No. 60975037).

References

1. Koller, D., Weber, J., Malik, J.: Robust multiple car tracking with occlusion reasoning. In: Eklundh, J.-O. (ed.) ECCV 1994. LNCS, vol. 801, pp. 189–196. Springer, Heidelberg (1994)

[3] http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/

2. Zhong, J., Sclaroff, S.: Segmenting foreground objects from a dynamic textured background via arobust kalman filter. In: IEEE International Conference on Computer Vision, pp. 44–50 (2003)
3. Oliver, N.M., Rosario, B., Pentland, A.: A bayesian computer vision system for modeling human interactions. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 831–843 (2000)
4. Zhong, J., Sclaroff, S.: Segmenting foreground objects from a dynamic textured background via a robust kalman filter. In: International Conference on Computer Vision, vol. 1, pp. 44–50. IEEE, Los Alamitos (2003)
5. Wren, C.R., Azarbayejani, A., Darrel, T., Pentland, A.: Real-time tracking of the human body. IEEE Transactions on Pattern Analysis and Machine Intelligence 19, 780–785 (1997)
6. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 246–252 (1999)
7. Elgammal, A., Harwood, D., Davis, L.: Non-parametric model for background subtraction. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1843, pp. 751–767. Springer, Heidelberg (2000)
8. Heikkila, M., Pietikainen, M.: A texture-based method for modeling the background and detecting moving objects. IEEE Transaction on Pattern Analysis and Machine Intelligence 28, 657–662 (2006)
9. Yao, J., Odobez, J.M.: Multi-layer background subtraction based on color and texture. In: IEEE Workshop on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
10. Liao, S., Zhao, G., Kellokumpu, V., Pietikainen, M., Li, S.Z.: Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes. In: IEEE Conference on Computer Vision and Pattern Recognition (2010)
11. Li, L., Huang, W., Gu, I.Y.H., Tian, Q.: Statistical modeling of complex backgrounds for foreground object detection. IEEE Transaction on Image Processing 13, 1459–1472 (2004)
12. Kim, K., Chalidabhongse, T., Harwood, D., Davis, L.: Real-time foreground-background segmentation using codebook model. Real-Time Imaging In Special Issue on Video Object Processing 11, 172–185 (2005)
13. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transaction on Pattern Analysis and Machine Intelligence 24, 971–987 (2002)
14. Ojala, T., Pietikainen, M., Harwood, D.: A comparative study of texture measures with classification based on feature distributions. Pattern Recognition 29, 51–59 (1996)
15. Koller, D., Weber, J., Huang, T., Malik, J., Ogasawara, G., Rao, B., Russell, S.: Towards robust automatic traffic scene analysis in real-time. In: IEEE International Conference on Pattern Recognition, vol. 1, pp. 126–131 (1994)
16. Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: Principles and practice of background maintenance. In: IEEE International Conference on Computer Vision, vol. 1, pp. 255–261 (1999)
17. Fisher, R.: The pets 2004 surveillance ground-truth datasets. In: IEEE International Workshop on Performance Evaluation of Tracking and Surveillance(PETS 2004), vol. 5, pp. 1–5 (2004)

Continuous Surface-Point Distributions for 3D Object Pose Estimation and Recognition

Renaud Detry and Justus Piater

University of Liège, Belgium
Renaud.Detry@ULg.ac.be, Justus.Piater@ULg.ac.be

Abstract. We present a 3D, probabilistic object-surface model, along with mechanisms for probabilistically integrating unregistered 2.5D views into the model, and for segmenting model instances in cluttered scenes. The object representation is a probabilistic expression of object parts through smooth surface-point distributions obtained by kernel density estimation on 3D point clouds. A multi-part, viewpoint-invariant model is learned incrementally from a set of roughly segmented, unregistered views, by sequentially registering and fusing the views with the incremental model. Registration is conducted by nonparametric inference of maximum-likelihood model parameters, using Metropolis–Hastings MCMC with simulated annealing. The learning of viewpoint-invariant models and the applicability of our method to pose estimation, object detection, and object recognition is demonstrated on 3D-scan data, providing qualitative, quantitative and comparative evaluations.

1 Introduction

Autonomous systems need to acquire object models for detection, recognition and manipulation. Models should be acquired autonomously, which implies a method that does not require precisely controlled environmental conditions, exact ground-truth pose, or full 360 viewpoint covering. Furthermore, partial models should be directly usable, and allow for incremental completion.

We present a 3D, probabilistic object-surface model, along with mechanisms for probabilistically integrating unregistered 2.5D views (range images) into the model, and for segmenting model instances in cluttered scenes.

Our model encodes object structure through continuous probability density functions representing the distribution of object-surface points. This allows us to achieve detection by probabilistic inference, effectively avoiding explicit model-to-scene correspondences. Our method learns an initial model from a single view of an object; the model can then be used to detect and estimate the pose of the object in novel scenes, provided that the view is sufficiently similar. If a new view provides more information, the model can be extended, in principle until the entire surface is completely modeled. Model learning and model exploitation are thus seamlessly integrated. We demonstrate that our approach is competitive with state-of-the-art methods. While performing at least as well as state-of-the-art algorithms on public datasets, our approach shows advantages

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 572–585, 2011.

in the paradigmatic and technical rigor of the techniques it builds on. Instead of defining e.g. probabilistic metrics on top of ad-hoc likelihood functions, our sensor model is intrinsically probabilistic. This approach allows for theoretical abstraction and flexibility. In particular, familiar building blocks from statistical learning are applied in appropriate places, such as kernel density estimation, Monte Carlo integration and inference, and expectation-maximization (see following sections).

Model learning is demonstrated on 3D-scan data from Biegelbauer and Vincze [1], and on the popular CAD dataset of Hetzel et al. [2]. The applicability of our method to pose estimation in cluttered point clouds is demonstrated on the data of Biegelbauer and Vincze, and object recognition rates are presented for the dataset of Hetzel et al.

2 Related Work

The modeling of objects from point-cloud data has been achieved through a variety of approaches [3]. The idea is generally to break down the object surface into a number of primitives; an object is then described by describing each primitive, and possibly by also describing their relative spatial configuration. Primitives correspond e.g. to complex parametric shapes such as superquadrics [1,4], local surface descriptor [2,5,6,7,8,9], or local edge descriptors [10]. In this work, primitives correspond directly to 3D points, with each point further parametrized by a local surface orientation computed from the distribution of the k nearest neighbors. In the following, we simply refer to these position-orientation pairs as *(5D) points*.

Depending on the application, recording the geometric structure of surface primitives, i.e. their relative spatial configuration, may or may not be necessary. Object recognition motivates discriminative models. Methods aiming at object recognition (without segmentation/pose estimation) [2,5,8] may completely ignore the spatial configuration of primitives, or encode it implicitly. They may also match a model by matching each view contained in the model separately, therefore also avoiding view registration. Conversely, pose estimation and segmentation require the modeling of the global shape of the object through the encoding of relative primitive configurations [6,7,9,10]. This generally leads to a generative model. Although it may not necessarily be their primary aim, generative models often provide recognition, too [7].

When building a generative 3D model from multiple views, it becomes necessary to derive an exhaustive registration of individual 2.5D views. Our method learns an initial model from a single view of an object; the model can then be used to detect and estimate the pose of the object in novel scenes, provided that the view is sufficiently similar. If a new view provides more information, the model can be extended, in principle until the entire surface is completely modeled. Thus, model learning and model exploitation are seamlessly integrated.

Detection and alignment of generative models is typically achieved through the matching of model descriptors to scene descriptors, possibly followed by

geometrically-constrained optimization [7,9]. We follow a different approach: We encode object structure through a continuous probability density function representing the distribution of object-surface 5D points. This allows us to achieve detection through probabilistic inference, which in turn avoids explicit model-to-scene correspondences. Our model is inferred by a Markov-chain Monte-Carlo (MCMC) algorithm which yields the maximum-likelihood pose of a model in an arbitrary scene.

Within a point-cloud reconstruction, the quantity of information conveyed by a point from a large and uniform surface is arguably smaller than the information conveyed by a point on a smaller, distinctive surface. In other words, the contribution of a surface segment to the identity of an object is generally not proportional to the number of points supporting the segment in a point-cloud reconstruction. Many 3D modeling techniques acknowledge this observation, e.g. through the detection of salient points [11], or the use of surface primitives of varying size [7]. We proceed by splitting object points into groups that represent object *parts* of different spatial size, and give each part the same weight in the detection process.

We note that the problem of 3D pose estimation (and recognition) has also been addressed for 2D images [12,13]. Image-based methods often rely on the matching of 2D patch descriptors; they work best on highly textured objects. Although these methods can be fast and convenient to deploy, their 3D estimates are generally less accurate than those obtained on range data.

This work is inspired by the work of Detry et al. [10], from which we borrow the idea of representing low-level sensor data with probability density functions. This paper goes beyond the work of Detry et al. in multiple ways. Inference is approached differently: we present a maximum likelihood MCMC algorithm, while Detry et al. compute a posterior density through belief propagation and importance sampling. Our learning method autonomously registers independent views, and autonomously identifies parts from spatial structure. Finally, our paper demonstrates the application of our system to range data, which is structurally different from the sparse-stereo edge data used by Detry et al., with two important results: (1) Our method is applicable to a much wider range of input data and does not depend on the heavy-weight ECV (Early Cognitive Vision) system. (2) Contrary to Detry et al., we can – and do – compare our results to competing approaches.

3 Object Model

As mentioned above, we consider point-cloud reconstructions in which each point is composed of a position and a local surface normal. The surface normal is computed at each point of the cloud from the covariance matrix C of its k nearest neighbors [14]. Let us denote by e_1, e_2, e_3 the eigenvalues of C, with $e_1 \geq e_2 \geq e_3$; depending on whether $e_1 - e_2$ is smaller or greater than $e_2 - e_3$, the local orientation is set to the eigenvector associated to e_3 or e_1 respectively, allowing for stable orientations on both surface and line configurations. Denoting

by S^2 the 2-sphere (the space of unit 3D vectors), computing local orientations yields a point-cloud $O = \left\{\left(\lambda^\ell, \theta^\ell\right)\right\}_{\ell\in[1,n]}$ where $\lambda^\ell \in \mathbb{R}^3$ and $\theta^\ell \in S^2$. We note that surface normals correspond to axial information; in other words, θ is equivalent to $-\theta$.

Our pose estimation method relies on the modeling of 3D surfaces with *surface-point distributions*. A surface-point distribution is a probability density function which models the spatial configuration of 5D points sampled from an object's surface. The function has a high value in regions surrounding object surfaces, and a low value elsewhere.

Surface-point distributions are represented with kernel density estimation. Kernel density estimation (KDE) allows one to model a continuous density function from a set of *observations* drawn from it, by assigning a local kernel function to each observation [15]; the density is estimated by summing all kernels. KDE allows us to define a continuous distribution from the point-cloud reconstruction of an object. The surface observations we are dealing with are points which belong to $\mathbb{R}^3 \times S^2$; denoting the separation of point components and kernel parameters into positions and orientations by $x = (\lambda, \theta)$, $\mu = (\mu_t, \mu_r)$, we define our kernel as

$$\mathbf{K}_{\mu,\sigma}(x) = \mathbf{N}_{\mu_t,\sigma_t}(\lambda)\, \mathbf{\Theta}_{\mu_r,\sigma_r}(\theta)\,, \tag{1}$$

where μ is the kernel mean point, $\sigma = (\sigma_t, \sigma_r)$ denotes the isotropic bandwidths in position and orientation, $\mathbf{N}$ is a trivariate isotropic Gaussian kernel, and $\mathbf{\Theta}$ corresponds to a pair of antipodal S^2 von-Mises Fisher distributions. The S^2 von-Mises Fisher distribution corresponds to a Gaussian-like distribution on 3D unit vectors [16]. Formally, the value of $\mathbf{\Theta}$ is given by

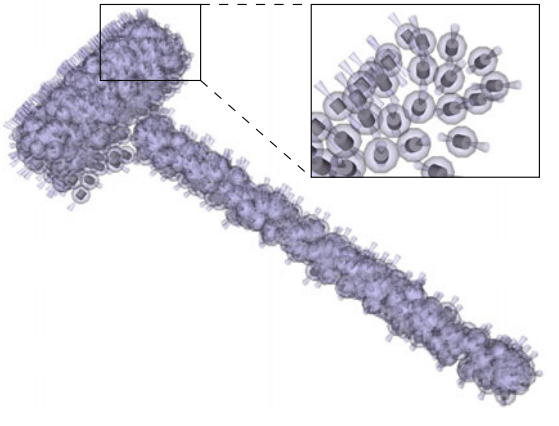


Fig. 1. Surface-point distribution computed from the 3D scan of a mallet. Surface-point observations are rendered with cylinders. The axis of a cylinder represents the orientation of the associated observation. Kernels are illustrated with translucent shapes: spheres and cones show one standard deviation in position and orientation respectively.

$$\mathbf{\Theta}_{\mu_r,\sigma_r}(\theta) = C_3(\sigma_r)\frac{e^{\sigma_r\,\mu_r^T\theta} + e^{-\sigma_r\,\mu_r^T\theta}}{2}, \tag{2}$$

where $C_3(\sigma_r)$ is a normalizing constant. The pair of antipodal von-Mises Fisher kernels in $\mathbf{\Theta}$ models the lack of direction in surface-normal orientations (see above); hence $\mathbf{K}_{(\lambda,\theta),\sigma}(x) = \mathbf{K}_{(\lambda,-\theta),\sigma}(x)$. The bandwidths σ_t and σ_r are computed using a k-nearest neighbor technique [15] on point positions. A surface observed through a point-cloud $\{x^\ell\}_{\ell\in[1,n]}$ is modeled with $\psi(x) = \sum_{\ell=1}^{n} \mathbf{K}_{x^\ell,\sigma}(x)$. An illustration is provided in Fig. 1.

We model an object composed of q parts with a set of surface-point distributions $\{\psi_i(x)\}_{i\in[1,q]}$; each surface distribution $\psi_i(x)$ models the distributions of points belonging to part i. All parts are defined in a common reference frame, so that $\sum_{i=1}^{q} \psi_i(x)$ yields a reconstruction of the whole object.

4 Inference

Model detection relies on the observation that a surface-point distribution can readily be used as a "3D template" that provides an *object pose likelihood* when convolved with the surface-point distribution of a scene. Let us consider an object model $\{\psi_i(x)\}_{i\in[1,q]}$. Also, let us denote by $SE(3) = \mathbb{R}^3 \times SO(3)$ the group of 3D poses, and by $\phi(x)$ the surface-point distribution of an arbitrary scene in which the object appears. We model the pose of the object with a random variable $W \in SE(3)$; the distribution of the object's pose in the scene modeled by $\phi(x)$ is given by

$$p(w) \propto \prod_{i=1}^{q} m_i(w), \tag{3}$$

with

$$m_i(w) = \int \psi_i(x)\,\phi(t_w(x))\,\mathrm{d}x, \tag{4}$$

where $t_w(\cdot)$ denotes a rigid transformation by w. Each integral $m_i(w)$ corresponds to the evaluation at w of the cross-correlation of part i with the scene.

Pose estimation is achieved by searching for the maximum of $p(w)$. Furthermore, the value of $p(w)$ at its peak may be used as a matching score, hence yielding object detection and recognition (see Section 6.2).

As computing the maximum-likelihood (ML) object pose $\arg\max_w p(w)$ is analytically intractable, we approximate it with Monte Carlo methods. The integrals $m_i(w)$ are approximated as:

$$m_i(w) \simeq \frac{1}{n}\sum_{\ell=1}^{n} \phi(t_w(x_\ell)) \quad \text{where} \quad x_\ell \sim \psi_i(x). \tag{5}$$

Simulating $p(w)$ directly is not possible, although simulating a random variate w_* from one integral $m_i(w)$ can be achieved as follows:

1. Generate $x_\phi \sim \phi(x)$,
2. Generate $x_\psi \sim \psi_i(x)$,
3. Generate $w_* \sim f(w)$, where $f(w) = \phi(t_w(x_\psi))$, by selecting w_* from a uniform distribution on the transformations which map x_ψ onto x_ϕ.

The ML pose $\arg\max_w p(w)$ is computed via simulated annealing on a Markov chain whose invariant distribution at iteration j is proportional to $p^{1/T_j}(w)$ [17,18], where T_j is a decreasing cooling schedule such that $\lim_{j\to\infty} T_j = 0$. The chain is defined with a mixture of two local- and global-proposal Metropolis–Hastings transition kernels, which are detailed below. Our choosing of the standard Metropolis–Hastings algorithm is motivated by the complexity of $\mathbb{R}^3 \times S^2$, which renders the calculation of local derivatives difficult. Also, $p(w)$ is likely to present a large number of narrow modes. A mixture of global and local proposals will compromise between distributed exploration of the pose space and fine tuning of promising regions. The independence-chain component of our transition kernel requires a global proposal function which we can simulate, and which should ideally resemble $p(w)$. In this paper, the global proposal corresponds to $s(w) = \sum_i m_i(w)$, which can be simulated by selecting $i \sim U_{[1,\ldots,q]}$, and sampling from $m_i(w)$. The local proposal for the random-walk component of the transition kernel is given by the $SE(3)$ kernel

$$\mathbf{K}^\star_{\mu,\sigma}(x) = \mathbf{N}_{\mu_t,\sigma_t}(\lambda)\, \mathbf{\Theta}^\star_{\mu_r,\sigma_r}(\theta)\,, \tag{6}$$

where $\mathbf{\Theta}^\star$ is a pair of antipodal S^3 von-Mises Fisher distributions, and rotations θ and μ_r are expressed as quaternions [19]. The location bandwidth σ_t of this kernel is set to a fraction of the size of the object, which in turn is computed as the standard deviation of input object points to their center of gravity. Its orientation bandwidth is set to a constant allowing for 5° of deviation. The complete algorithm is listed in Fig. 2. For our purposes, the mixture weight ν is typically set to 0.75; T_0 and T_N are set to 0.5 and 0.05 respectively; N is of the order of 5000. Simulated annealing does not guarantee convergence to the global maximum of $p(w)$. Hence, we run several chains in parallel and eventually select the best estimate.

The model presented above is intrinsically sensible to relative spatial resolution within the input point cloud: the cross-correlation of parts with scene evidence (3) will be proportional to the global value scale of $\phi(x)$ in the region covered by the model. Unfortunately, the spatial resolution of 3D scans is generally not uniform. For example, objects closer to the sensor will generate more return than further ones. Hence, the maximum of the pose likelihood (3) may not correspond to the pose that best explains *surface shape*. In the experiments presented below, we largely mitigate this effect by evaluating scene densities $\phi(x)$ as the *maximum* of underlying kernel evaluations at x. We note that model distributions $\psi_i(x)$ are not concerned by this issue since they are integrated over multiple views.

Finally, we note that the expression of $p(w)$ can be identified to an application of the Belief Propagation algorithm to a Markov random tree. The tree root W models the object pose. All the other nodes of the tree are leaves, which we

Initialize w_0 arbitrarily
Initialize σ_t and σ_r as explained in the text
For $j = 0$ to N :
$T_j = \max \left\{ T_0 \left(\frac{T_N}{T_0} \right)^{j/N}, T_N \right\}$
Sample $u \sim U_{[0,1]}$
If $u < \nu$:
Sample $w_* \sim s(w)$ *(global proposal)*
Sample $v \sim U_{[0,1]}$
If $v < \min \left\{ 1, \left(\frac{p(w_*)}{p(w_j)} \right)^{1/T_j} \frac{s(w_j)}{s(w_*)} \right\}$: $w_{j+1} = w_*$
Else : $w_{j+1} = w_j$
Else :
Sample $w_* \sim \mathbf{K}^{\star}_{w_j,(\sigma_t,\sigma_r)}(w)$ *(local proposal)*
Sample $v \sim U_{[0,1]}$
If $v < \min \left\{ 1, \left(\frac{p(w_*)}{p(w_j)} \right)^{1/T_j} \right\}$: $w_{j+1} = w_*$
Else : $w_{j+1} = w_j$

Fig. 2. Simulated annealing algorithm

denote by X_i. The network compatibility potential linking W to X_i is defined by $\psi_i(t_w^{-1}(x))$, where $t_w^{-1}(\cdot)$ denotes the inverse of a transformation by w, such that $(t_w \circ t_w^{-1})(x) = x$ for all x in $\mathbb{R}^3 \times S^2$. Observation potentials are given by $\phi(x)$. Each integral $m_i(w)$ corresponds to the message sent from X_i to W in a belief-propagation inference of the marginal distribution $p(w)$.

5 Learning

The generation of a model from a single point-cloud reconstruction of an object is described in Section 5.1. Section 5.2 explains how a model is learned from multiple views.

5.1 Modeling a Point-Cloud Reconstruction

Learning a model from a point-cloud reconstruction amounts to identifying the number and shape of object parts. Object parts are computed by clustering object points in $\mathbb{R}^3$; they are identified through the mixture of k trivariate Gaussians that best explains object point positions. K mixtures of $q = 1, ..., K$ Gaussians are fit using the Expectation-Maximization (EM) algorithm, and the most appropriate mixture is selected in a way inspired by the Bayesian information criterion [20]: the selected mixture is the one that minimizes

$$-2 \log L + Cq \log n, \tag{7}$$

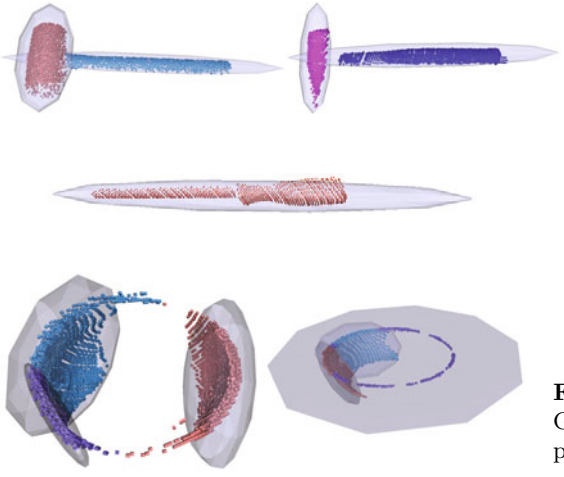


Fig. 3. Objects and their parts. Color indicates to which part a point belongs. Ellipsoids illustrate the mixture components that identify object parts.

Fig. 4. Example pose estimates. Grey dots correspond to scene points; color dots show object points aligned to the scene through pose estimation. The single-part model fails to produce a correct estimate (top), whereas a two-part model succeeds (bottom).

where L is the maximized value of the data likelihood, n is the number of points, and C is a numerical constant; the difference between the Bayesian information criterion and Eq. 7 is in the additional factor C which allows us to strongly penalize large mixtures, hence keeping the number of parts reasonably low. The object model M is eventually composed of q surface-point distributions $\psi_i(x)$ built through KDE on the points that belong to cluster i.

Clustering is responsible for the identification of characteristic object parts (Fig. 3), drawing attention to smaller areas that would otherwise be overwhelmed by larger surfaces. In Fig. 4, the top image shows a pose estimate computed from a single-part model of a hammer. Inference finds a best match of the handle of the hammer on a screwdriver, and ignores the unmatched head of the hammer. In the bottom image, the two-part model of Fig. 3 draws inference towards a correct estimate. The part identification method described above is similar to the procedure of Bouchard and Triggs [21], except that their work eventually expresses parts as cluster centers. Here, clustering is exclusively used to *identify* parts. Parts are represented by fine-grained surface-point densities (Section 3), which hold significantly more information than a single Gaussian.

5.2 Learning from Multiple Views

The construction of a model that expresses the full 3D geometry of an object requires pairwise registration of multiple views. Naturally, only pairs of sufficiently overlapping views can be registered. Finding overlapping views through an exhaustive registration of all pairs is unfortunately rather inefficient. Therefore, a meta-process should ideally detect strongly correlated views, which are good

candidates for registration [7]. In this section, we present a somewhat simpler method, which iteratively integrates views into a model, expecting each additional view to overlap with at least one of the previous views. Let us assume that each view contains n points, and let us denote by M_ℓ a model made up of ℓ views, and denote by O_ℓ the set of points used to construct M_ℓ. The first model M_1 is built, following the procedure of the previous section, from the points produced by the first available view v_1. Let us then assume that we have a model M_ℓ constructed from ℓ views, and the set O_ℓ from which it was built. Adding $v_{\ell+1}$ to the model works as follows. The pose of M_ℓ is estimated in $v_{\ell+1}$ (Section 4), which allows us to transform the points of $v_{\ell+1}$ into the object reference frame, yielding an object-registered point set T. A set of points $O_{\ell+1}$ that spans $\ell+1$ views is then formed by selecting $n/(\ell+1)$ points at random from T and $n\ell/(\ell+1)$ points from O_ℓ. $M_{\ell+1}$ is constructed by applying the procedure of Section 5.1 to $O_{\ell+1}$.

6 Evaluation

In the following experiments, models typically contain 1 to 4 parts. Scene surface-point distributions are computed from 5000 scan points. In order to limit the computational cost of inference, the total number of surface-point observations within object parts is limited to 500. The number of parallel chains in MCMC inference is typically set to 16. Our implementation estimates the pose of a model in a scene in about 5–10s on an 8-core desktop computer, and its memory footprint is always below 50MB. The cost of detecting multiple objects is linear in the number of objects.

6.1 Cluttered-Scene Pose Estimation

The suitability of our model for pose estimation in cluttered scenes is demonstrated on 3D-scan data from Biegelbauer and Vincze [1]. We learned a model of a mallet, a hammer, a screwdriver, and two bowls, using between 1 and 4 segmented range views of each object. The objects and their parts are illustrated in Fig. 3. The pose of these objects was estimated in 4 range scenes. Because points from the ground plane represent approximately 85% of each scene, we removed these prior to detecting the objects, by isolating them through RANSAC plane fitting. Although this step is not necessary, it significantly lowers inference time; also, through this process, we put our system in the same experimental conditions as Biegelbauer and Vincze. As illustrated in Fig. 5, all 11 pose estimates were correct. We followed the scenario of Biegelbauer and Vincze and reproduced the experiment several times using different software random seeds, and every run lead to the same correct estimates. When using models made of a single part, instead of the multi-part models of Fig. 3, only 7 out of 11 poses were correct, for reasons identical to these explained in Section 5.1. Despite its simplicity, the part-learning process is instrumental in discriminating between objects of similar shapes.

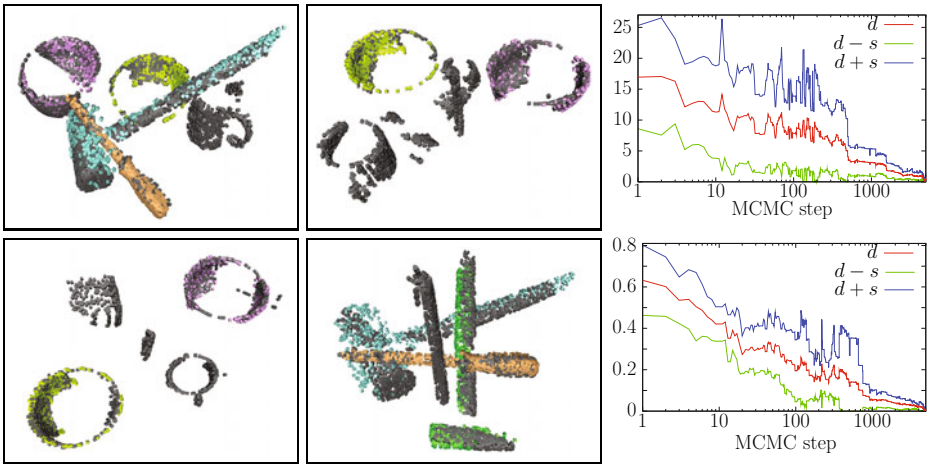


Fig. 5. Left side: cluttered scenes with pose estimates. Grey dots correspond to scene points. The rest of the dots illustrate pose estimates: they correspond to object points (Fig. 3) aligned to the poses of the objects in the scenes. There are 11 pose estimates (4 in the top-left scene, etc.). Right side: convergence of the MCMC process, as the mean position distance to optimum (top-right, distance in mm) and mean orientation distance to optimum (bottom-right, distance in radians) (see text for details).

The two graphs of Fig. 5 illustrate MCMC convergence. Pose estimation consists in running multiple Markov chains in parallel. The convergence of a pose estimation process e can be illustrated by tracking, at each MCMC step i, the distance d_i^e between the position component of the ground truth pose and the position component of the chain which is, at step i, the closest to the ground truth pose. The graph in the top-right corner of Fig. 5 (red curve) shows, as a function of the MCMC step i, the average $d_i = \frac{1}{11}\sum_e d_i^e$ across the eleven pose estimates of Fig. 5 (we note that the i axis is in log scale). The green and blue curves indicate an interval of one standard deviation s_i, with $s_i = \sqrt{\frac{1}{11}\sum_e (d_i^e - d_i)^2}$. The bottom-right graph illustrates orientation convergence. We note that as we do not have ground truth poses for the scenes of Fig. 5, the eleven ML estimates shown in the figure were used as ground truth. For both position and orientation, the error and error variance rapidly decrease between steps 1 and 1000, then smoothly converge to zero.

We note that the scenes of Fig. 5 contain more objects than those shown in Fig. 3. These objects are not part of the experiment simply because we do not have segmented views of them. A segmented view of the deeper bowl (purple in Fig. 5) was also missing. Its model was built from data extracted from the top-left scene of Fig. 5. It seemed relevant to include that object because of its similarity with the second (yellow) bowl.

6.2 Object Detection and Recognition

As mentioned above, the value of the pose-likelihood expression (3) at its peak may be used as a matching score, hence yielding object detection and recognition.

Object detection was evaluated on the online-available CAD dataset of Hetzel et al. [2]. This dataset contains 258 simulated range images for each of its 30 objects (Fig. 6). It is divided into a training and a testing set, containing respectively 66 and 192 views of each object. We learned a view-invariant model of each object using its 66 training views, providing them to the multi-view learning algorithm of Section 5.2 in the order in which they appear on the dataset website. Even though this order is not always ideal, it allowed for the construction of a good model of most objects. Fig. 7 shows eight examples. The bunny and the dinosaur are correctly reconstructed. The deodorant bottle is missing a side; this is explained by the symmetry of the object, which causes all views to be registered to the same side of the model.

Object detection determines whether a given model is present in a view. Detecting an object in a view amounts to estimating the object's ML pose w in that view (Section 4), reading the value of the pose distribution at w, and comparing the "score" $p(w)$ to a threshold (the whole process taking 5–10s). Thresholds were learned as follows:

1. We instantiated all 30 object models in 300 views of the training set – 10 views from each object – providing 9000 training scores. Fig. 8 (dashed curves) shows the distribution of the resulting scores for two objects.
2. For each object o, we trained a binary naive Bayes classifier on the 300 training scores produced by o, providing means of distinguishing o from all other objects.

Object detection rates were obtained by instantiating (Section 4) the 30 models in 300 images from the testing set, yielding 9000 testing scores. Fig. 8 (solid curves) shows the distribution of the resulting scores for two objects. Confronting the testing scores to the 30 detection classifiers yielded a 98% detection rate, i.e. out of the 9000 binary classifications, there are 298 true positives, 8580 true negatives, 2 false negatives and 120 false positives.

By contrast to object detection, object recognition determines, given one view, which object this view is most likely to show. For this purpose, we trained a single SVM classifier on the 9000 training scores obtained above. This classifier allows us to determine which object an arbitrary image corresponds to, by matching (Section 4) all 30 object models to that image, and submitting the 30 resulting scores to the classifier. On a set of 300 images from the testing set of the database, we obtained a 99% recognition rate, i.e. 297 true positives and 3 false positives. This result is directly comparable and competitive with recent discriminative approaches on the same dataset which yield 98% [8] and 93% [2]. It is also comparable to the 95% presented in Section 8.1 of the article from Mian et al. [7], although the object library used by Mian et al. is a superset of the one we are using. Recognizing which object a view belongs to requires the inference of all known object models; recognition is linear in the number of object models.

Fig. 6. Object library from Hetzel et al. [2]. Illustration kindly provided by Li and Guskov [8].

Fig. 7. Object points obtained from the registration of sequences of 66 views. Color indicates learned parts; objects on the first row are made up of a single part, whereas those on the second row yield two- or three-part models. The models of the deodorant bottle (top-right) and of the pedal (bottom-right) were not correctly constructed, because of symmetries and similarities within the objects.

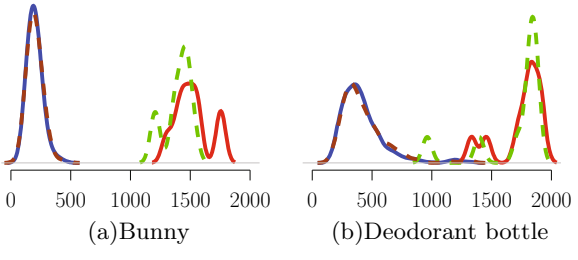


Fig. 8. Distribution of detection scores for the bunny and deodorant bottle (see Fig. 7). The dashed green line shows the distribution of the scores resulting from the instantiation of the object model in all training images of that object. The dashed brown line (almost overlapping with the blue line) corresponds to the instantiation of the model in the rest of the training set. The red line corresponds to testing images of the object; the blue line corresponds to testing images of the other objects. The scores provided by the bunny model are clearly separable. The deodorant bottle is less robustly detected, largely because of a second, similar bottle in the object set.

We emphasize that the classifiers above are only applied to find appropriate score-separating thresholds or planes. The underlying inference mechanism is not discriminative, and goes further than object recognition by providing an $SE(3)$ object pose.

7 Conclusion

We presented the definition, inference and construction of a 3D object model. The model consists of a set of parts represented with smooth surface-point densities. Object pose likelihood is defined through the cross-correlation of parts with scene evidence. The ML pose is computed through simulated annealing on a Markov chain whose invariant distribution is proportional to an increasing power of the pose likelihood, yielding an effective balance between exploration and convergence. The learning procedure probabilistically registers and fuses partly overlapping object views and identifies object parts through expectation-maximization. The suitability of our model for pose estimation was demonstrated on cluttered range scenes, using a set of objects of similar shapes; object recognition results competitive with recent generative and discriminative methods were obtained on a publicly available dataset.

While performing at least as well as state-of-the-art algorithms on public datasets, our approach shows advantages in paradigmatic and technical rigor of the techniques it builds on. Instead of defining e.g. probabilistic metrics on top of ad-hoc likelihood functions, our sensor model is intrinsically probabilistic. This approach allows for theoretical abstraction and flexibility. In particular, familiar building blocks from statistical learning are applied in appropriate places, such as kernel density estimation, Monte Carlo integration and inference, and expectation-maximization. Using rigorous, formal building blocks also facilitates the adaptation of the system to different situations. For instance, for problems where local derivatives of the pose density are available, using hybrid Monte Carlo instead of Metropolis-Hastings would improve inference performances.

Acknowledgments. The authors warmly thank Dr. Georg Biegelbauer and Pr. Markus Vincze for providing us with 3D-scan data. This work was supported by the Belgian National Fund for Scientific Research (FNRS) and the EU Cognitive Systems project PACO-PLUS (IST-FP6-IP-027657).

References

1. Biegelbauer, G., Vincze, M.: Efficient 3D object detection by fitting superquadrics to range image data for robot's object manipulation. In: IEEE International Conference on Robotics and Automation (2007)
2. Hetzel, G., Leibe, B., Levi, P., Schiele, B.: 3D object recognition from range images using local feature histograms. In: Computer Vision and Pattern Recognition, pp. 394–399 (2001)
3. Campbell, R.J., Flynn, P.J.: A survey of free-form object representation and recognition techniques. Comput. Vis. Image Underst. 81, 166–210 (2001)

4. Solina, F., Bajcsy, R.: Recovery of parametric models from range images: The case for superquadrics with global deformations. IEEE Trans. Pattern Anal. Mach. Intell. 12, 131–147 (1990)
5. Frome, A., Huber, D., Kolluri, R., Bülow, 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)
6. Johnson, A.E., Hebert, M.: Using spin images for efficient object recognition in cluttered 3D scenes. IEEE Trans. Pattern Anal. Mach. Intell. 21, 433–449 (1999)
7. Mian, A.S., Bennamoun, M., Owens, R.A.: Three-dimensional model-based object recognition and segmentation in cluttered scenes. IEEE Trans. Pattern Anal. Mach. Intell. 28, 1584–1601 (2006)
8. Li, X., Guskov, I.: 3D object recognition from range images using pyramid matching. In: International Conference on Computer Vision, pp. 1–6 (2007)
9. Rusu, R., Blodow, N., Beetz, M.: Fast point feature histograms (FPFH) for 3D registration. In: IEEE International Conference on Robotics and Automation (2009)
10. Detry, R., Pugeault, N., Piater, J.: A probabilistic framework for 3D visual object representation. IEEE Trans. Pattern Anal. Mach. Intell. 31, 1790–1803 (2009)
11. Li, X., Guskov, I.: Multi-scale features for approximate alignment of point-based surfaces. In: Eurographics Symposium on Geometry Processing (2005)
12. Rothganger, F., Lazebnik, S., Schmid, C., Ponce, J.: 3D object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. Int. J. Comput. Vision 66, 231–259 (2006)
13. Collet, A., Berenson, D., Srinivasa, S., Ferguson, D.: Object recognition and full pose registration from a single image for robotic manipulation. In: IEEE International Conference on Robotics and Automation (2009)
14. Liang, P., Todhunter, J.S.: Representation and recognition of surface shapes in range images: A differential geometry approach. Computer Vision, Graphics, and Image Processing 52, 78–109 (1990)
15. Silverman, B.W.: Density Estimation for Statistics and Data Analysis. Chapman & Hall/CRC, Boca Raton (1986)
16. Fisher, R.A.: Dispersion on a sphere. Proc. Roy. Soc. London Ser. A (1953)
17. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science 220, 671–680 (1983)
18. Andrieu, C., de Freitas, N., Doucet, A., Jordan, M.I.: An introduction to MCMC for machine learning. Machine Learning 50, 5–43 (2003)
19. Sudderth, E.B.: Graphical models for visual object recognition and tracking. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA (2006)
20. Schwarz, G.: Estimating the dimension of a model. The Annals of Statistics 6, 461–464 (1978)
21. Bouchard, G., Triggs, B.: Hierarchical part-based visual object categorization. In: Computer Vision and Pattern Recognition, vol. 1, pp. 710–715 (2005)

Efficient Structured Support Vector Regression

Ke Jia[1,2], Lei Wang[1], and Nianjun Liu[1,2]

[1] College of Engineering & Computer Science, The Australian National University
[2] National ICT Australia (NICTA), Canberra, Australia
{ke.jia,lei.wang,nianjunl}@cecs.anu.edu.au

Abstract. Support Vector Regression (SVR) has been a long standing problem in machine learning, and gains its popularity on various computer vision tasks. In this paper, we propose a structured support vector regression framework by extending the max-margin principle to incorporate spatial correlations among neighboring pixels. The objective function in our framework considers both label information and pairwise features, helping to achieve better cross-smoothing over neighboring nodes. With the bundle method, we effectively reduce the number of constraints and alleviate the adverse effect of outliers, leading to an efficient and robust learning algorithm. Moreover, we conduct a thorough analysis for the loss function used in structured regression, and provide a principled approach for defining proper loss functions and deriving the corresponding solvers to find the most violated constraint. We demonstrate that our method outperforms the state-of-the-art regression approaches on various testbeds of synthetic images and real-world scenes.

1 Introduction

Structured prediction has recently attracted much attention and many approaches have been developed. Structured learning studies the problems in which both inputs and outputs are structured and exhibit strong internal correlations. It is formulated as the learning of complex functional dependencies between multivariate input and output representations. Structured learning has significant impact in addressing important computer vision tasks including image denoising [1], stereo [2], segmentation [3,4,5], object localization [6,7], human pose estimation [8,9], to name a few. A popular approach is to generalize from the max-margin binary/multiclass classification problems to incorporate structured information [10,11,5]. On the other hand, there still lacks a fundamental way to deal with structured prediction from regression viewpoint, which we believe is potentially a more general approach. The reason mainly comes from the continuous value range of regression outputs, where infinite possible labels exist. It is therefore difficult to be parameterize-modeled, efficiently trained and predicted. In the structured regression field, Weston *et al.* proposes a linear map model in [12] to unify support vector classification and regression, and develops a methodology to solve high dimension problems using joint kernel maps. The joint kernel based structured prediction method is also utilized in [7], which performs well to localize objects on real images. An alternative structured regression

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 586–598, 2011.

approach other than joint kernel is proposed in [9], in which output correlations are modeled in and the outputs act as auxiliary features.

However, although [12] tries to generalize support vector classification and regression and succeeds in independent training scenario, it fails in structured regression. The prediction on data in the form of Wx, which is formulated in [12], does not correctly count in label dependencies. Because feature map x does not depend on the label outputs in regression, no label correlation is modeled in the prediction. In fact, the "structured" term in [12] for regression can be regarded as a simple combination of node and neighboring local features. In addition, a fixed label loss function is given in [12], which makes the penalty of label discrepancy unchangeable for its specializations, both classification and regression. The dependency among label outputs is properly formulated in the projection function of [9], but it only considers the internal correlations among outputs. In this case, the label smoothness is applied equally to all nodes belonging to the same output and those having significant outputs. Obviously, valuable information in context is not fully utilized here to assist the smoothing. The work in [9] uses the original SVR constraints in their structured regression framework. The structured learning usually deals with data in very large scale. In this case, the SVR settings in [9] will accumulate quite a number of constraints. Consequently, it significantly increases the computational cost.

In this paper, we propose to address the problem of structured prediction from regression viewpoint. In particular, we have following three contributions. First, we devise a projection function that takes the label output as an additional weight for the pairwise feature. By doing this, our projection contains a full set of dependencies including three kinds of correlation between output variables, input variables and input-output interrelated variables. Second, by utilizing bundle method learning process, our algorithm efficiently solves the complex objective function in a small number of iterations. By further adopting the 1-slack trick, the number of constraints increases by only 1 in each iteration. A smaller sized constraint set significantly speeds up the training process on large scale data. This setting of constraints also improves the robustness of our algorithm, because it alleviates the impact of outliers. Third, we design a principled approach to define proper loss functions for tasks with various regression targets, and also to derive corresponding solvers to find most violated constraint with respect to the defined loss function. This provides an effective way for practitioners to design suitable loss functions for a given task. Focusing on our study case, we attain an appropriate loss function and derive an efficient solver following our principled approach. We empirically evaluate our approach on both synthetic and real-world data sets, and it outperforms the state-of-the-art regression methods.

The outline of the paper is as follows. The proposed approach is described in Section 2, in which the detail of discussion about loss function and most violated constraint is also included. Comparison of our approach to related methods including M^3Ns, Joint Kernel Maps and $SOAR_{svr}$ is also presented in Section 2. Section 3 reports the experimental results on synthetic and real data sets, together with the empirical analysis. A conclusion is drawn in Section 4.

2 Our Approach

2.1 Problem Description

Let us denote an image instance as $X \in \mathcal{X}$ and its observation as $Y \in \mathcal{Y}$. They are defined over a graph $G = (V, E)$ of size $|V| = d$, respectively. Y is a continuous space defined over its pixels with each pixel assigned a real value $y_i \in \mathcal{R}$, and $Y = (y_1, y_2, \cdots, y_d)^T$. More specifically, let $i \in V$ index a node i and $ij \in E$ index an edge between vertexes i and j of the graph G.

In the training phase we have access to a set of T ground-truth images as $\{(X_t, Y_t)_{t=1}^T\}$. Our aim is to learn a W-parameterized projection function $F : \{X, Y, W\} \rightarrow Y'$. Here we need the model to take the correlations between output variables into account, *i.e.,* in the projection function, observation Y plays a role of an auxiliary for visual features to generate the global label outputs. The model parameter W is learned over the training set of T images. We require the optimal output of F for X_t to be as close as possible to its ground-truth value Y_t. Meanwhile, we need the model to generalize well on unseen images. In our work, for each node i, we assume that the projection F can be locally modeled by a linear function

$$f_i : \{X, Y, W\} \rightarrow y_i' \equiv f_i(x_i, \mathbf{y}^{-i}, W) = \langle w_v, x_i \rangle + \frac{1}{N} \sum_{j \in \mathcal{N}_i} \langle w_e, x_{ij} \rangle \, y_j, \qquad (1)$$

where $\mathbf{y}^{-i}$ is the $(d-1)$-dimensional output vector without the i-th entry, x_i and x_{ij} (or w_v and w_e) are local node and edge components of X (or W), and $\mathcal{N}_i$ denotes the set of the N neighboring nodes of the i-th node. Denoting $\phi(x_i, \mathbf{y}^{-i}) = (x_i, \frac{1}{N} \sum_{j \in \mathcal{N}_i} x_{ij} y_j)$, we have $f_i(x_i, \mathbf{y}^{-i}, W) = \phi(x_i, \mathbf{y}^{-i}) W$, where $W = (w_v^T, w_e^T)^T$. At the image level, we write the features in a matrix form as $\Phi(X, Y) = \{\phi(x_i, \mathbf{y}^{-i})|_{i=1}^d\}$. Thus, the projection function F can be expressed as

$$F(X, Y, W) = \{f_i(x_i, \mathbf{y}^{-i}, W)|_{i=1}^d\} = \Phi(X, Y) W. \qquad (2)$$

Now, given an unseen image X, this regression problem can be formally described as predicting the graph output Y^* by minimizing a loss function defined between an output Y and the projection upon it,

$$Y^* = \underset{Y \in \mathcal{Y}}{\operatorname{argmin}} \; \mathcal{L}_I(Y, F(X, Y, W)) = \underset{Y \in \mathcal{Y}}{\operatorname{argmin}} \; \sum_{i=1}^{d} L(y_i, f_i(x_i, \mathbf{y}^{-i}, W)).$$

The loss function $L(a, b)$ evaluates the difference between two labels a and b. It returns a positive value when $a \neq b$, and zero otherwise. Evidently, the minimum value of $\mathcal{L}_I$ should be zero, and this implies that the optimal output Y^* needs to satisfy $F(X, Y^*, W) = Y^*$. In terms of L_2-norm, it gives

$$\begin{aligned} Y^* &= \arg_{Y \in \mathcal{Y}}(F(X, Y, W) = Y) \\ &= \underset{Y \in \mathcal{Y}}{\operatorname{argmin}} \; \|F(X, Y, W) - Y\|^2. \end{aligned} \qquad (3)$$

Notice that the minimum value of zero can always be achieved when $F(X, Y, W)$ is a linear function of Y. We will show this later.

2.2 Our Max-Margin Formulation

Given an image-label pair (X_t, Y_t), we would like the Euclidean distance between the predicted label, $F(X_t, Y_t, W)$, and the ground-truth Y_t to be minimum for any candidate output Y. This yields, for the set of T training images,

$$\|F(X_t, Y_t, W) - Y\|^2 - \|F(X_t, Y_t, W) - Y_t\|^2 \geq \Delta(Y_t, Y) \ \ \forall t, Y. \tag{4}$$

where $t = 1, 2 \cdots, T$. Here $\Delta(Y_t, Y)$ denotes another loss function, which plays the role of *margin* between the true output Y_t and any other candidate output Y. This loss function could in general be an arbitrary function defined over the graph that measures the discrepancy between two label assignments. In other words, it is non-negative, symmetric, and attains zero if $Y_t = Y$. For the structured support vector regression, we need to avoid using the loss function like $\Delta(Y_t, Y) = \|Y - Y_t\|^2$. A comprehensive discussion about the $\Delta(Y_t, Y)$ will be given in Section 2.3.

By invoking (2), we expand left-hand-side (LHS) of (4) and after some algebra, it gives

$$2(Y_t^T - Y^T)\Phi(X_t, Y_t)W + Y^T Y - Y_t^T Y_t \geq \Delta(Y_t, Y) \ \ \forall t, Y. \tag{5}$$

Notice that the quadratic terms of W in the LHS of (4) have been canceled out, and this gives a linear function of W.

Now, we optimize (5) for all possible labels Y, and at the same time minimize the norm of W to avoid trivial solutions. Adding the slack variables $\xi_t \geq 0$ to account for violations, the optimization problem reads, for $\eta > 0$,

$$\begin{aligned} &\min_{W, \xi_t} \quad \frac{\|W\|^2}{2} + \frac{\eta}{T} \sum_{t=1}^{T} \xi_t \\ &s.t. \quad 2(Y^T - Y_t^T)\Phi(X_t, Y_t)W + {Y_t}^T Y_t - Y^T Y + \Delta(Y_t, Y) \leq \xi_t \ \ \forall t, Y. \end{aligned} \tag{6}$$

2.3 Parameter Learning and Inference Details

Bundle Method. For the optimization problem in (6), it has been shown in [13] that one may use a bundle method to find an approximate solution in polynomial time. For the convenience of analysis, we rewrite the constrains of (6) into the form of (4). The constraint related to the t-th image can thus be equivalently expressed as

$$\xi_t \geq \max_Y [\|F(X_t, Y_t, W) - Y_t\|^2 - \|F(X_t, Y_t, W) - Y\|^2 + \Delta(Y_t, Y)].$$

That is, the violations of constraint (RHS) is upper bounded by ξ_t (LHS). Given the current W, the bundle method can be used to optimize the objective function, which needs to identify the most violated constraint. Noticing that only the

last two terms depend on Y, this leads to efficiently solve the following column generation problem

$$\begin{aligned} Y_m &= \underset{Y \in \mathcal{Y}}{\operatorname{argmin}} \ [\|F(X_t, Y_t, W) - Y\|^2 - \Delta(Y_t, Y)] \\ &= \underset{Y \in \mathcal{Y}}{\operatorname{argmin}} \ [\|Y_t' - Y\|^2 - \Delta(Y_t, Y)], \end{aligned} \tag{7}$$

where $Y_t' = F(X_t, Y_t, W)$ given current approximated W.

The bundle method used in our training procedure (Algorithm 1) is guaranteed to approach the optimal solution to arbitrary precision in less than $O(\frac{1}{\zeta})$ iterations where ζ is the small tolerance in stopping criterion. By further adopting the one-slack trick of [13], empirically it always converges in a small number of iterations, which promises a tightly-bounded size for the constraint set $\mathcal{S}$ in our algorithm.

Loss Function and Most Violated Constraint. Now, we solve the column generation problem in (7) for the most violated constraint as follows. This part has the following contributions. Firstly, we develop a general approach to define proper loss functions and derive the corresponding solver to find the most violated constraint. Secondly, focusing on the special case of ε-insensitive loss, we derive a serial of possible solvers to efficiently identify the most violated constraint, the formulation in [12] is shown to be a special example in the serial.

In the space of $\mathcal{R}^d$, there exists at least one 2D affine plane on which the labels Y_t', Y_t and an arbitrary label assignment Y forms a triangle as shown in Fig. 1. For ease of exploration, denote $\|Y_t' - Y_t\| = a$, $\|Y - Y_t\| = b$ and $\|Y - Y_t'\| = c$, the angle $\angle Y Y_t Y_t' = \theta$. Thus we have $c^2 = a^2 + b^2 - 2ab\cos\theta$.

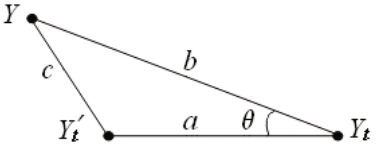


Fig. 1. The labels Y, Y_t' and Y_t in the triangular locations on a Euclidean plane

Assume that the label loss is a function of b, denoted as $\Delta(b)$. We can rewrite the most violated constraint (7) as:

$$\begin{aligned} Y_m &= \underset{b,\theta}{\operatorname{argmin}} \ [c^2 - \Delta(b)] \\ &= \underset{b,\theta}{\operatorname{argmin}} \ [a^2 + b^2 - 2ab\cos\theta - \Delta(b)]. \end{aligned}$$

Denoting $g(b, \theta) = a^2 + b^2 - 2ab\cos\theta - \Delta(b)$, we can get $\frac{\partial g}{\partial \cos\theta} = -2ab < 0$ because of $a > 0$ and $b > 0$, which means smaller g will always be obtained from

larger $\cos\theta$. Thus, the minimum g should come from $\cos\theta = 1$, that is $\theta = 0$. Therefore $c^2 = a^2 + b^2 - 2ab = (b-a)^2$ and also $g(b) = (b-a)^2 - \Delta(b)$.

Let's consider the partial derivation of g with respect to b:

$$\frac{\partial g}{\partial b} = 2b - 2a - \frac{\partial \Delta(b)}{\partial b}. \tag{8}$$

If the loss function is defined as those like $\Delta(Y_t, Y) = \|Y - Y_t\|^2 = b^2$, it will lead to $\frac{\partial g}{\partial b} = -2a < 0$. This makes $g(b)$ monotonically decrease with the increasing value of b. As a result, the minimum of $g(b)$ cannot be effectively achieved and this prevents the most violated constraint Y_m from being identified. Such a case should be avoided.

We can also rewrite (4) into the triangular form as

$$\begin{aligned} c^2 - a^2 \geq \Delta(b) \quad &\Rightarrow \quad (b-a)^2 - a^2 \geq \Delta(b) \\ &\Rightarrow \quad a \leq \frac{b^2 - \Delta(b)}{2b}. \end{aligned} \tag{9}$$

In regression, a number of different loss functions could be utilized. Here we focus on the commonly used ε-insensitive squared loss, and the other loss functions can be analyzed in a similar way. By invoking the ε-insensitive loss defined as $a \leq \varepsilon$, we would like to find out b and $\Delta(b)$ satisfying

$$\frac{b^2 - \Delta(b)}{2b} = \varepsilon \quad \Rightarrow \quad \Delta(b) = b^2 - 2b\varepsilon. \tag{10}$$

By combining (10) and (8), we obtain the partial derivation $\frac{\partial g}{\partial b} = 2\varepsilon - 2a > 0$. Therefore, when $b \in [\lambda, \infty)$, where λ is a positive constant, the optimal solution of the most violated constraint would be obtained at the point $b = \lambda$.

Because $\Delta(b) \geq 0$, b should be bounded at least $b \geq 2\varepsilon$ according to (10). For simplification, we choose $b \in [3\varepsilon, \infty)$ and work out the smallest $g(b)$ at $b = 3\varepsilon$. Thus, we have $\Delta(Y_t, Y_m) = \Delta(b) = 3\varepsilon^2$. Recall that $\theta = 0$, the most violated constraint Y_m can be calculated given Y_t' and Y_t,

$$Y_m = Y_t + 3\varepsilon \frac{Y_t' - Y_t}{\|Y_t' - Y_t\|}. \tag{11}$$

If choosing $b = 4\varepsilon$, we will come to $\Delta(b) = 8\varepsilon^2$. This is just the loss function that is directly defined and used in [12]. As shown, it is a special case of our definition. More importantly, our analysis explains when and why such a loss function and the alike would work.

Finally, the primal quadratic program (6) yields,

$$\begin{aligned} \min_{W,\xi} \quad & \frac{\|W\|^2}{2} + \eta\xi \\ s.t. \quad & \frac{1}{T}\sum_{t=1}^{T}[2(Y_m^T - Y_t^T)\Phi(X_t, Y_t)W + {Y_t}^T Y_t - Y_m^T Y_m + 3\varepsilon^2] \leq \xi, \\ & \xi \geq 0. \end{aligned} \tag{12}$$

Algorithm 1. Bundle Method Parameter Learning

Input: data X_t, labels Y_t, sample size T, insensitive radius ε, tolerance $\zeta > 0$
Initialize constraint set $\mathcal{S} \leftarrow \varnothing$, parameter $W \leftarrow \mathbf{0}$
repeat
  for $t = 1$ **to** T **do**
    $Y_t' \leftarrow F(X_t, Y_t, W)$
    $Y_m \leftarrow Y_t + 3\varepsilon \frac{Y_t' - Y_t}{\|Y_t' - Y_t\|}$
  end for
  Increase constraint set $\mathcal{S} \leftarrow \mathcal{S} \cup \{Y_m\}$
  $(W, \xi) \leftarrow$ Optimize (12) using all existing $Y_m \in \mathcal{S}$
until $\frac{1}{T}\sum_{t=1}^{T}[2(Y_m^T - Y_t^T)\Phi(X_t, Y_t)W + Y_t{}^T Y_t - Y_m^T Y_m + 3\varepsilon^2] \leq \xi + \zeta$

Inference of Y^*. According to (3), the optimal prediction can be calculated by solving the following matrix algebra:

$$\begin{aligned} & Y^* = F(X, Y^*, W) = X_v w_v + (\frac{1}{N}\sum_j X_e w_e) Y^* \\ \Rightarrow \quad & Y^* = (I - \frac{1}{N}\sum_j X_e w_e)^{-1} X_v w_v, \end{aligned} \tag{13}$$

where X_v and X_e are node and edge parts of standard feature map X.

Since the dimension of Y is generally high, which is the number of pixels in image processing. As a result, the inverse operation in (13) cannot be efficiently solved for a large-sized image. In this case, gradient-based optimizers can be used to efficiently attain an approximate solution, and we recommend to use unary outputs $X_v w_v$ as initialization.

2.4 Relationship to Existing Work

M^{3}Ns. The Max-Margin Markov Networks (M^{3}Ns) [11] is a structured version for SVM classification. We will show below that M^{3}Ns can be viewed as a special case of our SSVR framework.

For a structured classification, Y is a binary ("0" or "1") vector with the dimension of q^d, where q is number of categories for every node. There is one and only one "1" in Y, indicating one of the q^d possible label assignments for an image. The space of this Y is a special case of the continuous space $\mathcal{R}^d$ used in our SSVR, because $\mathcal{R}^d$ can be regarded as the same configuration vector with ∞^d dimension. As $Y^T Y = 1$ in the classification case, the constraint (5) is therefore

$$Y_t^T \Phi(X_t, Y_t) W - Y^T \Phi(X_t, Y_t) W \geq \Delta(Y_t, Y) \ \ \forall t, Y,$$

which is the same as that in M^{3}Ns. Here Y_t^T and Y^T can be regarded as a tensor to switch on only one column of $\Phi(X_t, Y_T)$ at one time, due to that there is only one "1" in Y and the others are all "0". Thus, M^{3}Ns is a special case of SSVR under conditions of discrete label space.

Joint Kernel Maps. As we described in Section 1, the objective function in [12], which takes the form of Wx, is commonly used in unary regression and structured classification algorithms, but it will fail and lost its "structured" sense in structured regression because it is just a simple combination of local node and edge features in this case. Our objective function in (1) correctly integrated the structured information, where the neighboring outputs will affect the result of central point as a smooth term.

Due to the modification of the objective function, the inference algorithm has to be changed in our framework to generate a global optimization of output assignments, while that in [12] just needs some simple linear algebra to get pointwise result. We adopted a gradient-based algorithm as the inference engine to calculate the global smoothed output.

SOAR$_{svr}$. Bo *et al.* proposed a closely related framework in [9], named SOAR$_{svr}$. SOAR is a rather general framework for structured regression. Our projection function in (1) can be regarded as a special case of that in SOAR. However, such specialization is necessary for a general framework like SOAR to be able to work for the image-based regression, for example, estimating the disparity value for each pixel in our work. The projection function in SOAR defines a parameterized weight for each component of a sample. When straightforwardly applied to a pixelwise image regression problem, SOAR will associate different weights to the locations of different pixels in an image. This will cause problems because the location of a given pixel changes with image translation, scaling, and rotation. To address this issue, the weights in our SSVR framework are designed to be independent of the pixel locations, which makes it able to handle the above image transformations and thus work for pixelwise image regression.

We also developed the projection function of SOAR into (1) by incorporating the pairwise features into the smooth term, as shown in the second term in (1). In contrast, only label outputs are taken into account in SOAR. This modification is important because the pairwise features, which measure the discrepancy between the features of neighboring pixels, will certainly provide more information about their similarity and thus help to achieve better cross-smoothing over pairwise pixels.

The learning algorithm in our approach is also different from that in SOAR. An original ε-insensitive SVR learning procedure is adopted in SOAR, while we utilize the bundle method in our algorithm, motivated by three main advantages. First, our max-margin formulation bounds the prediction Y in a tighter insensitive zone compared to the SVR formulation used in both unary SVR and SOAR. Recall that the dimension of Y is d. The original SVR applies 1D regression over each dimension of Y, and its insensitive zone is a hyper-cube with edge length of ε. A sample may not be penalized even if its distance from the origin is as far as $\sqrt{d}\varepsilon$. Differently, we bound the insensitive zone with a hyper-sphere of radius ε, which is a tighter zone inscribing into the hyper-cube. A sample

will surely be penalized once its distance from the origin is larger than ε. This insensitive zone can also be found in [14]. However, that work formulates the structured regression as a quadratic-constrained quadratic program, while our formulation solves the same problem in a quadratic program with linear constraints, which makes the learning process faster and lower the computational cost. Second, by utilizing the bundle method learning approach and adopting the one-slack trick, our algorithm finds the optimal solution in limited iterations with significantly less number of constraints than SOAR. In practice, our algorithm costs much less memory and is faster than SOAR for large-sized data sets. And the third, outliers can significantly affect the regression performance [15]. The existence of a small percentage of outliers is sometimes sufficient to make regression solvers produce very poor solutions. Real-world images have complex scene and always contain outliers that cannot be effectively explained by a regression model. Through identifying the most violated constraints, our approach uses a set of hyperplanes to approximate the objective function, forming a piecewise linear function. This avoids approximating the noise component in the objective function, and improves the robustness of the regression. This will be demonstrated by the experimental study.

To achieve a fair comparison between SOAR and our approach, we implement a SOAR-like algorithm (termed SSVR^{cube}), which utilizes the original SVR learning algorithm of SOAR, with the projection function modified by our formulation in (1) to deal with the pixelwise image regression tasks. We compare the results of both approaches in next section.

3 Experiments

Our approach has been evaluated on a variety of image testbeds. First we test on a synthetic binary images denoising data set, where the goal is to verify that the proposed approach indeed outperforms the state-of-the-art regression approaches. For data sets involving more complicated scenes, we show that our algorithm still outperforms them. During the experiments, we use the LibSVM package[1] for unary ε-SVR, and our approach is implemented in MATLAB 2006a.

3.1 Binary Synthetic Images

Our first goal in this section is to show the advantage of our approach over SVR by exploiting local pixel interactions, and also the improvement by utilizing bundle method learning procedure. We experiment with a binary denoising data set from [16]. The set contains 50 synthetic images corrupted with bimodel noises. Here all images are in the size of 64×64 pixels. 40 images are used in training, and they are divided into 4 equal groups (10 images in each group) to do the 4-folder tuning, where $\eta = 50$ is selected based on the tuning results. The rest 10 images are left aside for test.

[1] Available at http://www.csie.ntu.edu.tw/~cjlin/libsvm/

Fig. 2. Exemplar result of binary synthetic image denoising. From left to right: disturbed image, ground-truth and columns 3-5 are regression results of ε-SVR, SSVR^{cube} and our SSVR. Values lower than 0 or higher than 1 are adjusted to 0 or 1 respectively.

For a fair comparison, we use the same node features x_i for SVR, SSVR^{cube} and the proposed SSVR. The node features involving 2 dimensions which is the pixel grayscale value and 1-dim bias. We use the absolute difference of grayscale between the two neighboring pixels as the edge features x_{ij}, thus no additional information is provided to the structured learning approaches other than the unary one. The ground-truth label of pixels is either 0 or 1.

Table 1. Mean squared loss comparison of methods on the synthetic binary denoising dataset of [16]. The ε-SVR training time is roughly measured on LibSVM because the package does not record running time.

Methods	ε-SVR	SSVR^{cube}	SSVR
Training time (seconds)	7200	7.73	1.55
Mean squared loss	0.152	0.483	0.086

The experimental results in Fig. 2 and Table 1 validate that empirically our approach outperforms competition methods with a minimum mean squared loss of 0.086 and shortest training time. It is clear that our SSVR is more robust than other two algorithms. It can be observed that ε-SVR gives out a result almost the same as original disturbed image, and SSVR^{cube} failed completely on the task, due to the heavy outliers condition. Our approach successfully attains a proper model, which obviously adjustes the corrupted pixels by lifting noisy dark pixels (low values). Some bright pixels are inversely influenced for value reducing, because the learned model acts in a smoothness behavior. The training time of our algorithm is 1.55 seconds, and it labels an image in 0.54 seconds on average, measured by running on a desktop with 2.4GHz intel CPU and 2Gb memory.

3.2 Middlebury Stereo Datasets

To evaluate our approach on more complicated data sets, we test it on Middlebury Stereo Datasets from [17]. This data set consists of 6 scenes including *Art*, *Books*, *Dolls*, *Laundry*, *Moebius* and *Reindeer*. For each scene, 7 images are captured from different views (0 to 6), and 2 disparity maps related to view 1 and 5 are given as ground-truth. Images of scene *Laundry* and *Reindeer* are sized

Table 2. Comparison of methods on the disparity estimation. Fig. 3 displays some representative results on this dataset.

Methods	ε-SVR	SSVR^{cube}	SSVR
Training time (hours)	>200	1.65	1.29
Mean squared loss	n/a	5802.9	478.7

447 × 370 pixels, while others sized 463 × 370. We used the 12 images with ground-truth information available as our testbed, and two neighboring viewed images of each example are utilized for feature extraction. We left 3 images aside for test, and tuned the parameters using 3-folder cross-validation on 9 training images, the results suggest the parameter $\eta = 450$.

The ground-truth disparities are used as pixelwise labels, which value between 0 and 255. Two groups of features are adopted in our experiment, plus a 1-dim bias. First group is local visual features representing colors and textures, including 3-dim RGB color channels, 3-dim YCbCr color channels, and 11-dim texture features. For YCbCr color space, channel Y is image intensity, Cb and Cr are two color channels. Texture information is mostly contained in image intensity, so we applied the 9 Laws' masks [18] scaled 3 × 3 to channel Y. And the first Laws' mask is also applied to both color channels to extract haze. The local features include 17 dimensions in all. And we use 5-dim rough disparity estimations as features in the second group. For 5 different kinds of features (RGB color, YCbCr color, 4 Prewitt edge detectors oriented at 45° intervals filtered outputs, 3 × 3 Laws' mask response, and 5 × 5 Laws' mask response), we obtained the roughly approximated disparity maps respectively, using feature similarity. The disparity estimation of each pixel is attained by finding the most similar points along the same horizontal line in two neighboring viewed images. The top 2 similar points are chosen and average disparity of them is used as the estimation for current pixel. The 23-dimension node features plus 22-dimension edge features, which are absolute differences between neighbors, are input into different algorithms to regress the final disparities.

We compare our approach with other two methods, and results are shown in Fig. 3 and Table 2. The SSVR^{cube} result is scaled for better observation. LibSVM needs too much time to finish running, therefore its results is not available. All running times are measures on the same server with 2.8GHz CPU and 4Gb memory.

Our approach consistently obtains the lowest mean squared loss. The SSVR^{cube} algorithm, which uses the original SVR constraint set, is significantly influenced by outliers. The linear model it learned cannot properly represent the data distribution and highlight the outstanding features. Therefore, it over balances all labels to almost a constant, and the result keeps many intensity details, which plays an important role in features. Meanwhile, our approach demonstrates its robustness, and produces good result. On the large scale data set, the original SVR constraint set is extremely large, and thus it exponentially increases the complexity of quadratic program, running time for both ε-SVR and SSVR^{cube}

Fig. 3. Examples of disparity estimation on Middlebury stereo data sets. Columns from left to right: image, ground-truth, SSVR^{cube} result scaled for ease of view (original result is pretty dark), and prediction of our approach.

grow rapidly. Since our approach solves the problem iteratively with small constraint set, it takes the shortest time to learn the model. The average test time of one image is 13.94 seconds using a CSD gradient optimizer.

A more complicated feature space including 54 node features and 53 edge features was also experimented, the result of SSVR is similar to above one. But quadratic program in SSVR^{cube} cannot be solved because its redundant constraint set used up the memory, and LibSVM takes too long on calculation.

4 Conclusion

An efficient and robust structured support vector regression framework is proposed, and its performance is demonstrated through the problems of image denoising and disparity estimation in this paper. By incorporating the pairwise features into the projection function, our approach adaptively adjusts the impact of the neighboring pixels to the label of a given pixel according to their visual similarity. This advantage has been well demonstrated by the experiment on the binary synthetic data set. With the bundle method, our approach has significantly reduced the number of constraints by several orders since only the most violated ones are identified and used. As demonstrated by the experiment on the Middlebury stereo data set, our approach is superior to the existing methods on large-sized data sets in terms of both memory usage and running speed. Our analysis on the label loss function provides a principled way for practitioners to design suitable loss functions for a given task, which ensures the proper convergency of the bundle method learning process.

References

1. McAuley, J.J., Caetano, T.S., Smola, A.J., Franz, M.O.: Learning high-order mrf priors of color images. In: International Conference on Machine Learning (2006)
2. Carr, P., Hartley, R.: Minimizing energy functions on 4-connected lattices using elimination. In: International Conference on Computer Vision (2009)
3. Szummer, M., Kohli, P., Hoiem, D.: Learning cRFs using graph cuts. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 582–595. Springer, Heidelberg (2008)
4. Anguelov, D., Taskar, B., Chatalbashev, V., Koller, D., Gupta, D., Heitz, G., Ng, A.: Discriminative learning of markov random fields for segmentation of 3d scan data. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2005)
5. Taskar, B., Chatalbashev, V., Koller, D.: Learning associative markov networks. In: International Conference on Machine Learning (2004)
6. Ionescu, C., Bo, L., Sminchisescu, C.: Structural svm for visual localization and continuous state estimation. In: International Conference on Computer Vision (2009)
7. Blaschko, M., Lampert, C.: Learning to localize objects with structured output regression. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 2–15. Springer, Heidelberg (2008)
8. Kim, M., Pavlovic, V.: Dimensionality reduction using covariance operator inverse regression. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2008)
9. Bo, L., Sminchisescu, C.: Structured output-associative regression. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2009)
10. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large margin methods for structured and interdependent output variables. Journal of Machine Learning Research 6, 1453–1484 (2005)
11. Taskar, B., Guestrin, C., Koller, D.: Max-margin markov networks. In: Advances in Neural Information Processing Systems, vol. 16. MIT Press, Cambridge (2004)
12. Weston, J., Schölkopf, B., Bousquet, O.: Joint kernel maps. In: Cabestany, J., Prieto, A.G., Sandoval, F. (eds.) IWANN 2005. LNCS, vol. 3512, pp. 176–191. Springer, Heidelberg (2005)
13. Teo, C., Smola, A., Vishwanathan, S., Le, Q.: A scalable modular convex solver for regularized risk minimization. In: International Conference on Knowledge Discovery and Data Mining (2007)
14. Pérez-Cruz, F., Camps-Valls, G., Soria-Olivas, E., Pérez-Ruixo, J.J., Figueiras-Vidal, A.R., Artés-Rodríguez, A.: Multi-dimensional function approximation and regression estimation. In: Dorronsoro, J.R. (ed.) ICANN 2002. LNCS, vol. 2415, p. 757. Springer, Heidelberg (2002)
15. Colliez, J., Dufrenois, F., Hamad, D.: Robust regression and outlier detection with svr: Application to optic flow estimation. In: British Machine Vision Conference (2006)
16. Vishwanathan, S.V.N., Schraudolph, N.N., Schmidt, M.W., Murphy, K.P.: Accelerated training of conditional random fields with stochastic gradient methods. In: International Conference on Machine Learning (2006)
17. Hirschmller, H., Scharstein, D.: Evaluation of cost functions for stereo matching. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2007)
18. Davies, E.: Laws' texture energy in texture. In: Machine Vision: Theory, Algorithms, Practicalities, 2nd edn. Academic Press, San Diego (1997)

Cage-Based Tracking for Performance Animation

Yann Savoye and Jean-Sébastien Franco

INRIA Bordeaux University, France

Abstract. Full body performance capture is a promising emerging technology that has been intensively studied in Computer Graphics and Computer Vision over the last decade. Highly-detailed performance animations are easier to obtain using existing multiple views platforms, markerless capture and 3D laser scanner. In this paper, we investigate the feasibility of extracting optimal reduced animation parameters without requiring an underlying rigid kinematic structure. This paper explores the potential of introducing harmonic cage-based linear estimation and deformation as post-process of current performance capture techniques used in 3D time-varying scene capture technology. We propose the first algorithm for performing cage-based tracking across time for vision and virtual reality applications. The main advantages of our novel approach are its linear single pass estimation of the desired surface, easy-to-reuse output cage sequences and reduction in storage size of animations. Our results show that estimated parameters allow a sufficient silhouette-consistent generation of the enclosed mesh under sparse frame-to-frame animation constraints and large deformation.

1 Introduction

Modeling dynamic 3D scene across time from multiple calibrated views is a challenging problem that has gained full attention of the Computer Vision and Computer Graphics communities in recent years. Nevertheless, the relentless increase in demand of 3DTV industry has inspired researchers to exhibit new approaches able to produce reusable contents. Current pipelines try to achieve deformable mesh tracking using one or many linear or non-linear numerical optimizations. Video-based approaches are proven to be more convenient to acquire human performances. The field of targeted applications is very large, ranging from content generation for the 3D entertainment and motion picture industries, video game to sport analysis. Even if the human motion can be abstracted by the kinematic of rigid parts, the observed surface to track is purely non rigid because of small detail variations induced by clothes, hair motion and certain degrees of flexibility by virtue of natural properties.

A major challenge is to exhibit an efficient framework to achieve plausible boneless tracking that produces pleasing deformations, preserves the global appearance of the surface and offers flexible reusable outputs for animation. For this reason, we propose a new fully linear framework to track mesh models with full correspondence from multiple calibrated views.

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 599–612, 2011.

In this paper, we introduce a new cage-based mesh parametrization for tracked surfaces. The surface model, initially acquired by a laser or initial dense reconstruction method, is smoothly and volumetrically embedded in a coarse but topologically identical mesh, called *the cage*, whose vertices serve as control points. The resulting reduction in control parameters and space embedding yields an interesting new trade off to tackle the mesh tracking problem. More precisely, we focus on the estimation of desired enclosed models in a linear manner preserving the smoothness of the cage under sparse linear subspace constraints. Such constraints are defined over the enclosed mesh surface itself. We take advantage of optimal reduced parameters offered by the given coarse cage surrounding the surface. In order to avoid artifacts induced by the large number of degrees of freedom, the cage layer is enhanced with laplacian regularization. In other words, we embed to-be-tracked models in an adapted bounding polytope cage using generalized barycentric coordinates having local smooth properties, drawing inspiration from Computer Graphics animation-oriented model parametrizations.

The rest of the paper is organized in the following manner. Relevant works concerning markerless full body performance capture and non-rigid surface deformation are briefly reviewing and discussing in section 2. The problem statement and contributions are presented in section 3. Background and notations are introduced in section 4. We give an overview of our system in section 5. The core of the proposed method is detailed in section 6. Section 7 presents some results and evaluations of our novel technique in the context of multiple views performance animation. We show the effectiveness of our method with several examples. This paper is concluded in section 8 and limitations are discussed and 9.

2 Related Works

In this section, we will briefly describe relevant recent works in the field of 3D Video and Vision-based Graphics. We mainly focus on previous approaches addressing the problem of markerless full body performance capture from multiple views and interactive mesh deformation.

Performance Capture and 3D video. Markerless performance capture can be defined as the process of generating spatio-temporally coherent and connectivity preserving geometry. Recent years have seen strong interest for video-based performance capture dealing with video driven laser-scanned template or template-free deformation as well as skeleton-based or boneless tracking. One of the first pioneering work in surface capture for performance-based animation is proposed in [1] with a fully automated system to capture shape and appearance based on visual hull extraction, feature matching and dense reconstruction.

De Aguiar *et al.* have proposed in [2] a markerless approach to capture human performances from multi-view video that produces a novel dense and feature-rich output. This approach is enhanced in [3] by a novel optimization scheme for skeleton-based pose estimation and surface local estimation. Moreover Vlasic *et al.* present in [4] a framework for articulated mesh animation from multi-view

silhouettes. They also proposed a new method in [5] to dynamically perform shape capture using multi-view photometric stereo normal maps. Another approach presented in [6] introduces a tracking algorithm to realize markerless motion capture of skinned models in a four camera set-up using optical flow and skeletal subspace deformation into a nonlinear minimization framework.

3D Video processing becomes a promising visual media by enabling interactive viewpoint control. 3D video technologies are able to capture high-fidelity full 3D shape, motion and texture to offer free-and-rotate special effects. The full 3D video pipeline is presented in [7]. In the context of human motion synthesis from 3D video [8] proposed a method where surface motion graph is constructed to concatenate repeated motion.

Non-Rigid Deformation. High quality shape deformation has gained a lot of attention in recent years. For instance, non rigid mesh evolution can be performed using intrinsic surface deformation or extrinsic space deformation techniques.

A lot of effort has been done on surface-based deformation. There are several types of approaches exploiting a differential descriptor of the edited surface in terms of laplacian and differential coordinates for mesh editing [9,10]. Differential information as local intrinsic feature descriptors has been massively used for mesh editing in various frameworks over the decade [11,12,13]. Observing the local behaviour of the surface has been proposed recently in [14], where "as-rigid-as-possible" surface modeling is performed.

There has been a great deal of work done in the past on developing techniques for deforming a mesh with generalized barycentric coordinates. Inspired from the pioneering work presented in [15], cage-based methods are ideal for deforming a surface coherently by improving space deformation technique. The cage parametrization allows model vertices to be expressed as a linear combination of cage vertices for generating realistic deformation. This family of methods has important properties: quasi-conformal mappings, shape preservation and smoothness. To animate the model, cage vertices are displaced and the vertices of the model move accordingly through a linear weighted combination of cage geometry parameters. An approach to generalize mean value coordinates is introduced in [16]. The problem of designing and controlling volume deformations used to articulate characters are treated in [17], where the introduction of harmonic coordinates significantly improves the deformation stability thanks to a volumetric heat diffusion process respecting the connectivity of mesh volume.

A non linear coordinates system proposed in [18] called *Green Coordinates* leads space deformation with shape preserving properties. However such approaches require to obtain automatically a fairly coarse control mesh approximating enough a given surface [19,20]. Furthermore, there has been a great deal of work made feasible with respect to the work presented in [21,22,23], where the authors use an analogy to the traditional use of skeleton-based inverse kinematics.

3 Problem Statement and Contributions

Even if lots of methods reconstruct continuous surfaces from several viewpoints, we notice a lack of global, flexible and reusable parametrisation. However, finding suitable non-rigid performance animation model with reduced control parameters is a key problem in video-based mesh tracking. Unlike methods based on an underlying rigid skeleton, we aim to estimate the subspace deformation corresponding to the time-varying non-rigid surface evolution. Even if reconstruction is not part of our contribution, we deal with the peculiarities of surfaces that have been reconstructed from real video footage. Thus, we propose a new approach estimating an optimal set of cage vertices position allowing the template to fit the silhouettes, in a single-pass linear method ensuring cage smoothness under sparse subspace constraints.

Our method estimates cage parameters using animation sequence generated from calibrated multi-view image sequences and cage-based deformation. We exhibit an external parametrization involving a reduced number of parameters comparing to the number of enclosed mesh vertices. However, the key contribution is to solve a sparse linear system to estimate the best cage parameters reproducing the desired deformation of the enclosed model, given sparse constraints expressed on the enclosed model itself. This paper shows that cage parametrization can be used for video-based acquisition. To the best of our knowledge, this is the first approach using cage-based animation for performance capture.

4 Background and Notations

Multiple View Setup. We assume that a scene, composed of a single non-rigid object (Fig. 1), observed by a camera network composed of a set of i views where each of them corresponds to a fixed pinhole camera model. In addition, we assume that the static background can be learned in advance. The 4×3 projection matrix of the i^{th} camera is denoted by $\mathbf{P}_i$. The mapping from a 3D point $\mathbf{p} = (x, y, z)$ of the surface in the world coordinate space to its image coordinates (u_i, v_i) by projection in camera i is given by:

$$u_i(\mathbf{p}) = \frac{\mathbf{P}_i^1 \, (x, y, z, 1)^T}{\mathbf{P}_i^3 \, (x, y, z, 1)^T} \quad ; \quad v_i(\mathbf{p}) = \frac{\mathbf{P}_i^2 \, (x, y, z, 1)^T}{\mathbf{P}_i^3 \, (x, y, z, 1)^T} \tag{1}$$

where $\mathbf{P}_i^j$ is the j^{th} rows of P_i associated to the i^{th} camera. The projection matrix is obtained easily from the intrinsic and extrinsic camera. This projective formulation is used for qualitative evaluation.

Cage-Based Animation. In the rest of the paper, we use the following terminology. The coarse morphable bounding mesh $\mathcal{C}$ and the enclosed mesh $\mathcal{M}$ are respectively called *the cage* and *the model*. We assume that both entities are two-manifold triangular meshes fully-connected, with fixed toplogy, even if the model can be multi-component. The set of n cage vertices is denoted $\mathcal{V}_\mathcal{C} = \{c_1, \cdots, c_n\}$

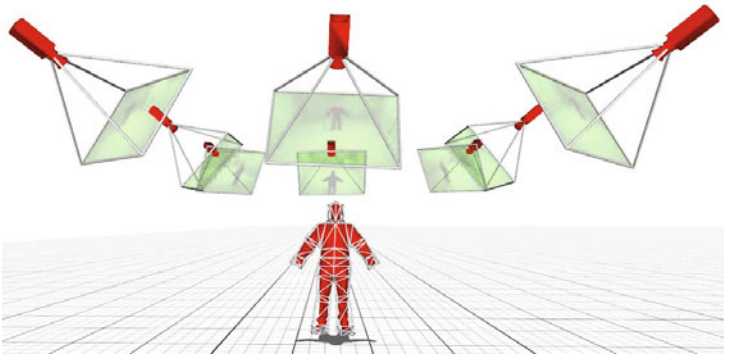


Fig. 1. Cage-based multiple-view setup

where c_k is the position coordinates of the k^{th} cage vertex, and the set of m model vertices with $\mathcal{V}_{\mathcal{M}} = \{v_1, \cdots, v_m\}$ where v_i is the location of the i^{th} model vertex.

5 Pipeline Overview

Before introducing technical details, we briefly describe the procedure of our method in an overview. Our method retrieves the video-based space warping of the observed surface across time. We assume that our framework exploits already reconstructed mesh sequences with full correspondence for the moment. We cast the problem as a minimization problem for cage recovery. The main challenge is to deal with the high number of degrees of freedom provided by the coarse cage.

As input of our pipeline (Fig. 2), we give a collection of images captured from the observed scene from calibrated views, and the reconstructed mesh sequence. As output, the system produces a sequence of cage's geometry parameters for each frame. In our system, we employ laplacian operator on the cage and harmonic coordinates to perform a space deformation surfacically regularized that allows us to obtain a coherent cage estimation. We estimate a sequence of cage parameters expressing the mesh at each animation frame using cage-based inverse kinematics process. The optimization process retrieves the pose in a least-squares sense from sparse motion constraints expressed directly over the enclosed

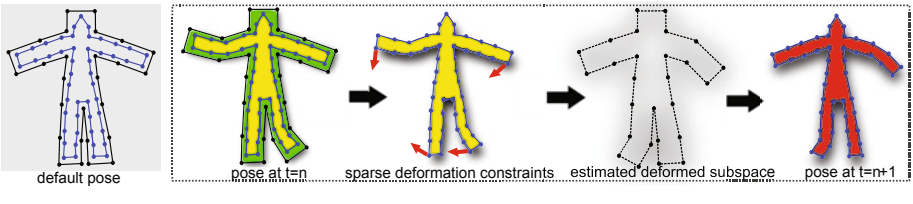


Fig. 2. Cage-based tracking: pipeline overview

surface and transfered into the subspace domain using its indirection. This new technique produces suitable outputs for animation compression and re-using.

6 Linear Least Squares Cage Fitting

This section presents the laplacian-based regularization applied exclusively on the cage structure, instead of current methods which regularize the full mesh itself. We also introduce the association of harmonic subspace deformation with cage-based dual laplacian to perform space tracking using a cage.

6.1 Harmonic Subspace Deformation

A cage is a coarse closed bounding polyhedral volume, not a lattice. This flexible low vertex-count polytope, topologically similar to the enclosed object, can efficiently control its deformation and produce realistic looking deformation. Model vertices are expressed as a linear combination of cage vertices. The weights are given by a set of generalized barycentric coordinates stored in a $m \times n$ deformation weight matrix denoted by $\mathcal{H}$. We also denote by $h_k\,(i)$ the normalized blend weights representing the deforming influence of the k^{th} cage vertex on the i^{th} model vertex. In another words, it is possible to deform an arbitrary point on the enclosed mesh expressed as a linear combination of the coarse mesh vertex position via a constant weight deformation. The cage-based forward kinematic function is expressed as follows:

$$v'_i = \sum_{k=1}^{n} h_k\,(i) \cdot c'_k \tag{2}$$

where v'_i is the deformed cartesian coordinates according to a vector of cage geometry $\{c'_1, \cdots, c'_n\}$. In order to produce as-local-as possible topological changes on the enclosed surface, the model is rigged to the cage using harmonic coordinates. The harmonic rigging is the pre-computed solution of Laplace's equation with Dirichlet boundary condition obtained by a volumetric heat diffusion in the cage interior volume. The resulting matrix corresponds to the matrix $\mathcal{H}$.

The subspace domain is the volume enclosed in the cage interior defined by control points. For each control point c_k, we define a harmonic function $h_k\,(\cdot)$ inside the cage by enforcing the interpolation property $h_k\,(c_j) = 1$, if $k = j$, and 0 if not, and asking that h_k be linear on the edges of the cage.

6.2 Laplacian-Based Subspace

Generally laplacian mesh editing techniques are interactive, but not real-time because the per-vertex laplacian is defined for the whole mesh and thus computationally expensive. A better idea to ensure the smoothness of the laser scan surface independently of its high-detail resolution is to define a Dual Laplacian operator. We propose to define the Laplacian operator on the cage instead of the

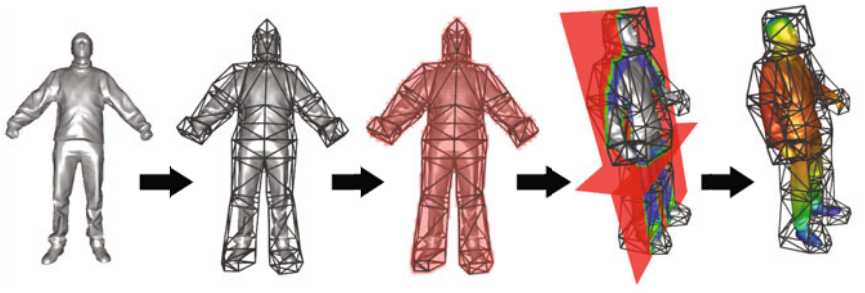


Fig. 3. Harmonic Rigging process: (from left to right hand side) laser scanned template, cage-based model embedding, cage voxelization, voxel tagging and harmonic weights computation

model to improve the computation process and to keep model detail properties good enough.

Let's denote by $\mathcal{L}_C(\cdot)$ the Dual Laplacian operator defined at each cage vertex domain by as the weighted sum of the difference vectors between the vertex and its adjacent neighbors. Given the fact that the cage is highly irregular, we prefer to use the cotangent weighing system in the computation of laplacian matrix. We also denote the differential coordinates of the cage by $\hat{\delta}$. Encoding each control vertex relatively to its neighborhood preserves the local geometry using differential coordinates.

6.3 Linear Subspace Constraints

The subspace topology is preserved across time because the default cage is seen as a connectivity mesh only and feature constraints are seen as external deformation. This surface-and-space based deformation technique preserves the mesh spatial coherence. The geometry of the cage can be reconstructed efficiently from its indirect harmonic coordinates and relative coordinates by solving a system of sparse linear equations. We cast the problem of deformation as a least-squares laplacian cage reconstruction process using a consistent minimization approach of an objective function. The cage parameters recover the sparse pose features by minimizing an objective function in a least-squares sense in order to fit a continuous volume. Then, the geometry of the desired model is simply obtained by generating its vertex position according to the expressed cage parameters obtained on the concept of least-squares cage.

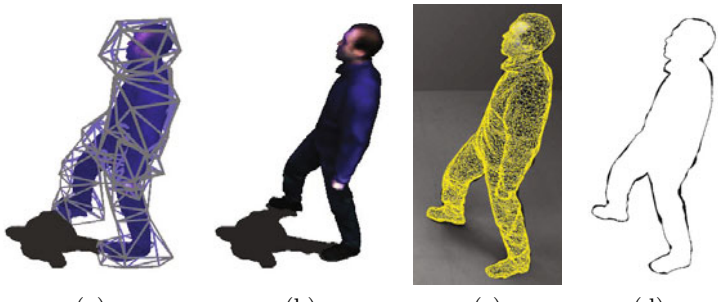
Given the differential coordinates, laplacian operator of the default cage, the harmonic weights $h_k(i)$ according to the cage and the model at the default pose, and several 3D sparse surface constraints, the absolute coordinates of the model geometry can be reconstructed by estimating the absolute coordinates of the cage geometry. The combination of the differential coordinates and harmonic coordinates allows the reconstruction of the surface by solving a linear system. We can formulate overall energy to lead an overdetermined linear system to extract the cage parameters.

This least-squares minimization problem can be expressed exclusively in term of cage geometry (for frame-to-frame animation) as demonstrated in the following formula:

$$\min_{c'_k} \left(\sum_{k=1}^{n} \left\| \mathcal{L}_C \left(c'_k \right) - \hat{\delta'}_k \right\|_2^2 + \sum_{v_i \in \mathcal{S}} \left\| v'_i - \sum_{k=1}^{n} c'_k \cdot h_k \left(i \right) \right\|_2^2 \right) \tag{3}$$

Note that the first term of the energy preserves the cage smoothness and ensures a pleasant deformation under sparse constraints. The space-based distortion energy is measured by the laplacian on the cage. The total local distortion measure for a deformation process is given by a quadratic energy term.

The second term of the energy enforces the position of vertices to fit the desired model defined by positional constraints. To our best knowledge, the simple global optimization component of our framework with such formulated constraints to minimize does not already exist in the literature. Overall energy performed by our technique reproduces harmonic space deformation recovery under indirected dual laplacian mesh editing. After the cage retrieval process, the geometry of



(a) (b) (c) (d)

Fig. 4. Qualitative evaluation of the cage-based reconstructed surface: (a) estimated cage, (b) textured enclosed surface, (c) cage-based model reprojected, (d) silhouette overlap error

(a) real image (b) 100% const. (c) 5% const. (d) 1% const.

Fig. 5. Influence of the number of positional constraints on the quality of enclosed mesh expressed by the estimate cage (the number of positional constraints are expressed in percentage of enclosed mesh vertices) (dataset courtesy of [4])

the desired enclosed model is reconstructed in linear combination function of cage geometry parameters related to the new estimation, preserving the fixed connectivity.

In order to deform the bounding cage, positional constraints are defined on the model using anchor points. We denote by v_i' the cartesian coordinates position of the target point at $t+1$ associated to v_i to create a positional constraint. $\mathcal{S}$ is the subset composed of a irregularly distributed collection of positional constraints (for each selected v_i that form $\mathcal{S}$) over the enclosed surface. The second term of the objective function measures how much the cage enforces sparse positional constraints. The transfer of surfacic constraints into the subspace domain exploiting the cage indirection is expressed by this energy term. In other words, the last formulation enables to express surface constraints directly in terms of cage parameters linearly using an inverse quasi-conformal harmonic mapping.

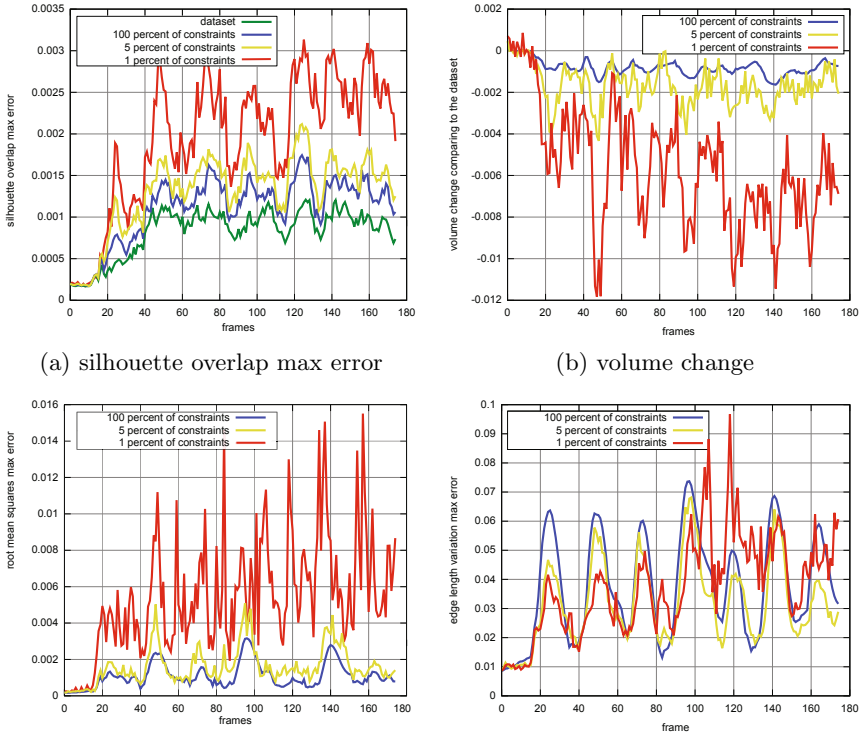
Given the control mesh for the previous frame in a deformation mesh sequence, we need to exploit frame-to-frame deformation of the finer mesh to automatically constructed an altered control mesh for every frames in the sequence. As shown on the results, the cage retrieval process only requires a small number of corresponding input vertices and their displacement to form sparse linear subspace constraints to be able to estimate a cage expressing a surface fitting to the silhouette good enough.

7 Results and Evaluation

This section describes our experiments using this system. Our framework, implemented with OpenGL and C/C++, proposes a robust mechanism to extract a cage for various potential applications. The entire process takes less than two seconds per frame without any code optimization, and uses solvers running on CPU. The algorithm performance is validated by both qualitative and quantitative evaluations. We show that the cage reproduces the 3D motion of life-like character accurately without requiring a dense mapping.

The performance of our system was tested on multi-view video sequences that were recorded with eight cameras at a resolution of 1600x1200 pixels. The template is composed of 10002 vertices and the cage is composed of 141 vertices (80% of parameter reduction of the enclosed model). To validate our method, some experimental results are shown on real datasets (Fig. 4). Qualitative evaluations are based upon visual comparisons (silhouette consistency) of each reconstructed frame with the original one and various percentage of vertex constraints randomly selected (Fig. 5). We also provide rendering feedback to allow qualitative evaluation on two different sequences with a total of 348 frames (Fig. 7).

We run our cage-based tracking method to measure how much the estimated cage-based deformation of the template can fit the observed silhouettes without applying an additional silhouette regularization on the enclosed surface. For our evaluation as shown on Fig. 6, we measure the fidelity of our output with several error metrics such as edge length variation, silhouette overlap error, root mean square volume variation comparing to the input dataset models.



(a) silhouette overlap max error

(b) volume change

(c) root mean square max error

(d) edge length max error

Fig. 6. Quantitative evaluation on MIT Crane dataset

We claim the feasibility of generating digital mesh animation for visualizing real scene of realistic human in motion using cage capture. In addition, the deformation driven by the cage offers an affordable silhouette-consistency with respects to all images recorded by all cameras. Because the fixed connectivity of the cage is preserved across the sequence our technique well suited for encoding and compression. To show the accuracy of cage-based tracking we have developed a 3D video player that displays in real-time the cage-based performance animation. To increase the realism of the rendering, the enclosed model is rendered using an omnidirectional texture mapping generated from the multiple views video stream. Cage-based deformation allows the 3D video player to produce a smooth and accurate frame-to-frame playback of time-varying mesh.

8 Discussion

We have shown the feasibility of using cage-based parametrization for multiple-views video-based acquisition. The main advantage of our framework is its linear form, as well as the reduction of mesh parameters, which is independent of the

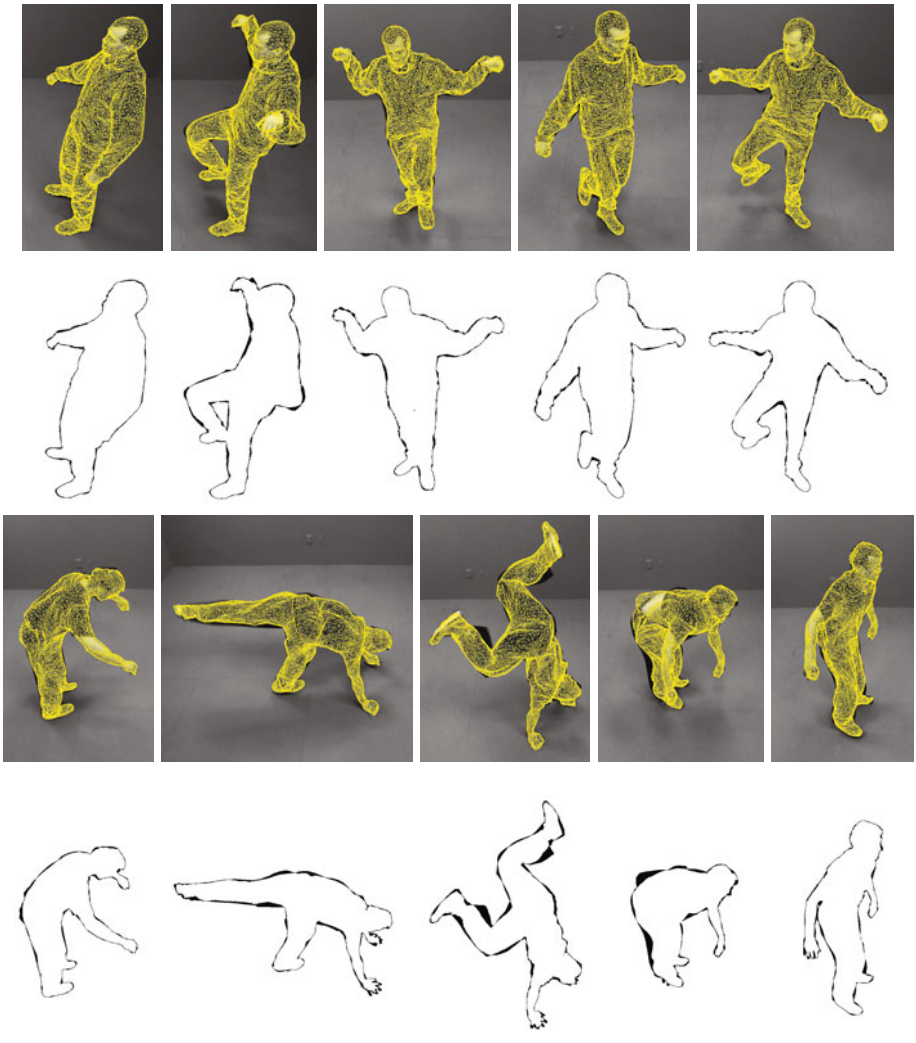


Fig. 7. Projection of the wireframe mesh generated by the estimated cage with 5% of constraints randomly selected (1^{st} and 3^{rd} rows), silhouette overlap between the rendered silhouette and the extracted silhouette (2^{nd} and 4^{th} rows.) (crane and handstand dataset).

surface resolution making possible reuse. Our harmonic cage-based deformation allows mesh rim vertices to fit the silhouette more precisely comparing to skeleton based deformation only because of more degrees of freedom. Our techniques can efficiently facilitated the extraction of the deformation subspace of mesh animations by retrieving the cage for all frames using a minimization framework fully independent of the model resolution. In addition, our technique drastically decreases the size of dataset without any lost of quality.

Nevertheless our method suffer of some drawbacks directly derived from laplacian and space deformation properties. For example, the volume shrinking can provoke interior folding and potential fold over under non-silhouette consistent target points. Another major limitation of our method is that the deformation result depend on the shape and the tessellation of the cage. Moreover automatic cage generation for the setup process is also an opened hard problem.

9 Conclusion and Future Works

Even if there has been seen a strong interest for template-based approaches and multi-view photometric for performance capture, no previous work tried to use cage-based parametrization for mesh sequence tracking and animation retrieval. In this paper, we have investigated the opportunities in-between cage-based tracking and multiple-views spatio-temporal reconstructed shape. We have developed a framework incorporating cage-based optimization in the context of the multi-view setup and captured the space deformation.

This cage-based deformation technique is a useful tool to improve the incremental reconstruction across time, because this method can provide a better control over the surface to allow rim vertices to fit the silhouette without prior knowledge of rigid parts. In this paper, we demonstrate the strength of harmonic coordinates used inside a linear minimization framework to reconstruct an enclosed mesh fitting the silhouette better than a skeleton. We show that our method is adapted in the context of a multiple-views setting with a proper experimental validation Finally, our novel approach is also very interesting for 3D video compression and animation reuse.

This work opens up a lot of new and interesting research directions. This algorithm is simple enough to be widely implemented and tested with previous framework. In the future, we plan to investigate and explore the possibility of achieving incremental 4D reconstruction, not relying on pre-shape dense sequences. Our method could be improved easily by integrating several image-based reconstruction cues such as sparse features like surface texture, silhouette and motion features observed in multiple viewpoint images.

References

1. Starck, J., Hilton, A.: Surface capture for performance-based animation. IEEE Comput. Graph. Appl. 27, 21–31 (2007)
2. de Aguiar, E., Stoll, C., Theobalt, C., Ahmed, N., Seidel, H.P., Thrun, S.: Performance capture from sparse multi-view video. In: SIGGRAPH 2008: ACM SIGGRAPH 2008 papers, pp. 1–10. ACM, New York (2008)
3. Gall, J., Stoll, C., de Aguiar, E., Theobalt, C., Rosenhahn, B., Seidel, H.P.: Motion capture using joint skeleton tracking and surface estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009), Miami, USA, pp. 1–8. IEEE Computer Society, Los Alamitos (2009)

4. Vlasic, D., Baran, I., Matusik, W., Popović, J.: Articulated mesh animation from multi-view silhouettes. In: SIGGRAPH 2008: ACM SIGGRAPH 2008 papers, pp. 1–9. ACM, New York (2008)
5. Vlasic, D., Peers, P., Baran, I., Debevec, P., Popović, J., Rusinkiewicz, S., Matusik, W.: Dynamic shape capture using multi-view photometric stereo. ACM Trans. Graphics (Proc. SIGGRAPH Asia) 28 (2009)
6. Ballan, L., Cortelazzo, G.M.: Marker-less motion capture of skinned models in a four camera set-up using optical flow and silhouettes. In: 3DPVT, Atlanta, GA, USA (2008)
7. Takai, T., Nobuhara, S., Yoshimoto, H., Matsuyama, T.: 3d video technologies: Capturing high fidelity full 3d shape, motion, and texture. In: International Workshop on Mixed Reality Technology for Filmmaking (in cooperation with ISMAR 2006) (2006)
8. Huang, P., Hilton, A., Starck, J.: Human motion synthesis from 3d video. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1478–1485 (2009)
9. Lipman, Y., Sorkine, O., Cohen-Or, D., Levin, D., Rössl, C., Seidel, H.P.: Differential coordinates for interactive mesh editing. In: Proceedings of Shape Modeling International, pp. 181–190. IEEE Computer Society Press, Los Alamitos (2004)
10. Sorkine, O.: Differential representations for mesh processing. Computer Graphics Forum 25, 789–807 (2006)
11. Lipman, Y., Sorkine, O., Alexa, M., Cohen-Or, D., Levin, D., Rössl, C., Seidel, H.P.: Laplacian framework for interactive mesh editing. International Journal of Shape Modeling (IJSM) 11, 43–61 (2005)
12. Au, O.K.C., Tai, C.L., Liu, L., Fu, H.: Dual laplacian editing for meshes. IEEE Transactions on Visualization and Computer Graphics 12, 386–395 (2006)
13. Luo, Q., Liu, B., Ma, Z.g., Zhang, H.b.: Mesh editing in roi with dual laplacian. In: CGIV 07: Proceedings of the Computer Graphics, Imaging and Visualisation, Washington, DC, USA, pp. 195–199. IEEE Computer Society, Los Alamitos (2007)
14. Sorkine, O., Alexa, M.: As-rigid-as-possible surface modeling. In: Proceedings of Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, pp. 109–116 (2007)
15. Meyer, M., Lee, H., Barr, A., Desbrun, M.: Generalized barycentric coordinates on irregular polygons. Journal of Graphics Tools 7, 13–22 (2002)
16. Ju, T., Schaefer, S., Warren, J.: Mean value coordinates for closed triangular meshes. In: SIGGRAPH 2005: ACM SIGGRAPH 2005 Papers, pp. 561–566. ACM, New York (2005)
17. Joshi, P., Meyer, M., DeRose, T., Green, B., Sanocki, T.: Harmonic coordinates for character articulation. ACM Trans. Graph. 26, 71 (2007)
18. Lipman, Y., Levin, D., Cohen-Or, D.: Green coordinates. In: ACM SIGGRAPH 2008 papers, pp. 78:1–78:10. ACM, New York (2008)
19. Ju, T., Zhou, Q.Y., van de Panne, M., Cohen-Or, D., Neumann, U.: Reusable skinning templates using cage-based deformations. ACM Trans. Graph. 27, 1–10 (2008)
20. Xian, C., Lin, H., Gao, S.: Automatic generation of coarse bounding cages from dense meshes. In: IEEE International Conference on Shape Modeling and Applications (Shape Modeling International 2009) (2009)

21. Sumner, R.W., Zwicker, M., Gotsman, C., Popović, J.: Mesh-based inverse kinematics. ACM Trans. Graph. 24, 488–495 (2005)
22. Der, K.G., Sumner, R.W., Popović, J.: Inverse kinematics for reduced deformable models. In: SIGGRAPH 2006: ACM SIGGRAPH 2006 Papers, pp. 1174–1179. ACM, New York (2006)
23. Shi, X., Zhou, K., Tong, Y., Desbrun, M., Bao, H., Guo, B.: Mesh puppetry: cascading optimization of mesh deformation with inverse kinematics. ACM Trans. Graph. 26, 81 (2007)

Modeling Dynamic Scenes Recorded with Freely Moving Cameras

Aparna Taneja, Luca Ballan, and Marc Pollefeys

ETH Zurich, Switzerland

Abstract. Dynamic scene modeling is a challenging problem in computer vision. Many techniques have been developed in the past to address such a problem but most of them focus on achieving accurate reconstructions in controlled environments, where the background and the lighting are known and the cameras are fixed and calibrated. Recent approaches have relaxed these requirements by applying these techniques to outdoor scenarios. The problem however becomes even harder when the cameras are allowed to move during the recording since no background color model can be easily inferred.

In this paper we propose a new approach to model dynamic scenes captured in outdoor environments with moving cameras. A probabilistic framework is proposed to deal with such a scenario and to provide a volumetric reconstruction of all the dynamic elements of the scene.

The proposed algorithm was tested on a publicly available dataset filmed outdoors with six moving cameras. A quantitative evaluation of the method was also performed on synthetic data. The obtained results demonstrated the effectiveness of the approach considering the complexity of the problem.

1 Introduction

Passive modeling of dynamic scenes is a challenging problem in computer vision. The aim is to recover a mathematical time-varying description of the scene using only videos recorded by some cameras. A considerable number of approaches have been developed in the past to address such a problem. Typically these techniques exploit the use of silhouette [1,2,3], color/stereo [4,5,6,7], shading [8,9] and motion [10] extracted from the videos in order to infer the geometry of the dynamic elements of the scene. In the case of silhouette based techniques, the geometry of the dynamic objects is recovered using either deterministic [1,11] or probabilistic [3,2] visual hull. Color information can be exploited by using either multi-view stereo [6,7] or narrow baseline stereo [4], or combining both together as proposed in [5]. Silhouette and color information can also be combined to improve the reconstruction results [12,13,14].

However, most of these works focus on controlled environments where the background is known or can be estimated and the cameras are fixed and calibrated. Only few approaches have tried to deal with outdoor scenarios mainly

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 613–626, 2011.

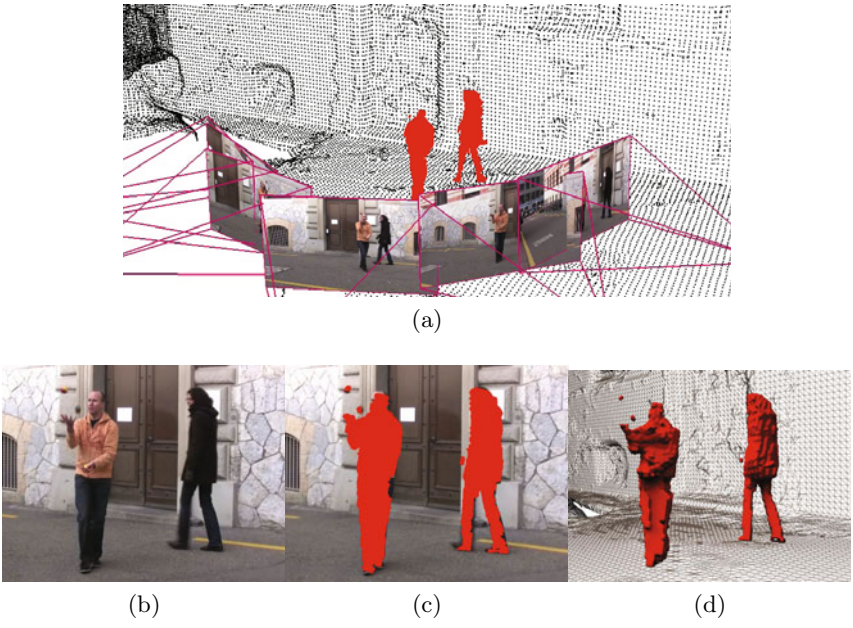


Fig. 1. Results obtained using our approach to model a dynamic scene with two people: one walking and the other juggling. (a) Volumetric reconstruction. (b) One frame of the videos used for the reconstruction. (c) Reconstructed volume projected back to the previous image. (d) Reconstruction rendered from another viewpoint.

resorting to some scene priors [15] or limiting the reconstruction quality at billboard level [16]. In particular, [15] performs on the assumption that a person is the only one dynamic element in the scene.

Unlike the above approaches, in this paper, we propose a technique to achieve full 3D reconstruction of the scene dynamics in outdoor uncontrolled environments filmed with moving cameras and without making any assumptions on the shape or the motion of elements to be reconstructed. In a sense, our approach can be considered similar, at least in principle, to a silhouette based approach.

Silhouette based approaches rely on the possibility of performing background subtraction on the entire video sequence. This is an easy task in a controlled environment but becomes hard in the more generic case of an outdoor scenario. In [2], for instance, the authors addressed such a scenario but assuming stationary cameras. The problem indeed, becomes even more challenging in the case of moving cameras since a per-pixel color model for the background cannot be recovered anymore. Some relevant works focusing on background subtraction have been developed in the recent past to address this kind of situations. However, these techniques resort to some priors on both the background and the foreground elements of the scene such as shape priors [17,18], color priors [16,19] and motion priors [20]. The first class assumes that the foreground objects can

only have specific shapes, for instance, human shapes. The second class assumes that the color models of the foreground and the background objects are known a priori. The last class instead makes assumptions on the type of motion of the dynamic elements. As an example, [20] assumes that the elements that are moving rigidly, with respect to camera, are background while the others are foreground.

In this paper we propose a technique to infer the geometry of the dynamic elements of a scene by exploiting the structural information of the static parts of the scene, which is inferred in a preprocessing stage, and the color information from the acquired video sequences.

To avoid building a per-pixel background color model from temporal video data for segmentation, we instead use the precomputed geometry of the static parts of the scene to transfer the current background appearance across multiple views. Given some images captured at the same time instant, our approach is based on projecting each image onto the other images and exploiting their differences. Something similar was partially exploited by [21] to achieve a deterministic and fast background subtraction of a person using three static cameras. These projections however generate some false detections which in our text will be referred as the *ghost* of the foreground (occlusion shadows in their paper). While in [21], the method simply eliminates these artifacts by intersecting all the reprojections, we instead exploit this information as well in a probabilistic framework. As described later in the paper, the ghost may help recover in some situations where no information can be obtained from the actual location of the dynamic object.

Since only the images captured at the same time instant are used to model the current scene we do not suffer from some issues that are common in background subtraction techniques such as changes in illumination or shadows.

Compared to the approach proposed in [16], where the authors suggest to retrieve the background color by exploring the temporal domain of each video independently, our approach exploits the spatial domain, retrieving this information from the other cameras at the same time instant. Moreover, in this approach we do not need an initial color model for the foreground which had to be specified by a user in [16].

This paper is organized in three parts. Section 2 describes the proposed reconstruction algorithm. Section 3 shows the experimental results. In the end, Section 4 draws the conclusions.

2 Reconstruction Procedure

The captured videos are first pre-processed in order to retrieve information about the cameras and the static elements of the scene. Subsequently, the geometries of all the dynamic elements of the scene are reconstructed. This section describes how the pre-processing stage is performed while the next section covers the reconstruction of the dynamic elements. For the sake of simplicity we refer to the static part of the scene as background and the dynamic part as foreground.

Structure-from-Motion (SfM) [22,23] and multi-view stereo [6] can be applied to some images of the scene, captured in absence of any dynamic elements, in

order to recover the background geometry. In our implementation we used the pipeline provided by Zach et al. [24] which generates a continuous mesh model for the background.

Each video camera is then calibrated both spatially and photometrically, and the video streams synchronized. To do so, we follow the approach described in [16]. More specifically, intrinsic parameters are recovered using [25] and they are assumed to be constant throughout the recording. Subsequently the pose, i.e., the extrinsic parameters, for each camera at each time instant, are computed with respect to the background geometry by matching the SIFT descriptors [26] extracted from the current frame and the SIFT descriptors previously extracted during the SfM procedure. These matches generate correspondences between 2D points in the current frame and 3D points in the background geometry. The pose of that camera at that specific time is recovered by applying the three points algorithm [27]. Temporal synchronization of the videos stream is performed using the corresponding audio streams as in [15].

Finally, the video streams are calibrated photometrically with respect to each other using the method proposed in [28]. More specifically a color transfer function mapping the color space of one camera into the color space of another is recovered for each pair of cameras. This is necessary to account for different settings in the cameras like different exposure time, gain and white balancing.

Our formulation is designed to estimate the 3D reconstruction of a single frame. For sake of simplicity, from here on, the analysis will focus only on a specific time instant t and the text will refer to images captured by the cameras as the images captured at that specific time t.

Let I_i denote the image captured by camera $i \in [1, .., n]$, and let π_i be the projection function mapping 3D points in the world coordinate system to 2D points in the image coordinate system of camera i according to both the intrinsic and the extrinsic parameters recovered during the previous stage.

Since both the background geometry and the projection function π_i are known, the depth map of the background geometry seen by camera i can be computed. Let's denote this depth map with Z_i. The value stored in each of its pixels represents the depth of the closest 3D point of the background geometry that projects to that pixel using π_i. In practice, Z_i can be easily computed in GPU by rendering the background geometry from the point of view of camera i and by extracting the resulting Z-buffer.

Given two cameras i and j, let R_{ij} denote the image obtained by projecting the image I_j into camera i, i.e., by rendering the background geometry from the point of view of camera i using the color information of camera j and taking into account the color transfer function between i and j. More formally, for each pixel p in R_{ij}, we know that $\pi_i^{-1}([p, Z_i^p]^T)$ represents the coordinates of the closest 3D point in the background geometry projecting in p. Note that π_i^{-1} is the inverse of the projection function π_i where the depth is assumed to be known and equal to Z_i^p. Therefore, the coordinates of pixel p in the image j are equal to

$$\pi_j(\pi_i^{-1}([p, Z_i^p]^T)) \tag{1}$$

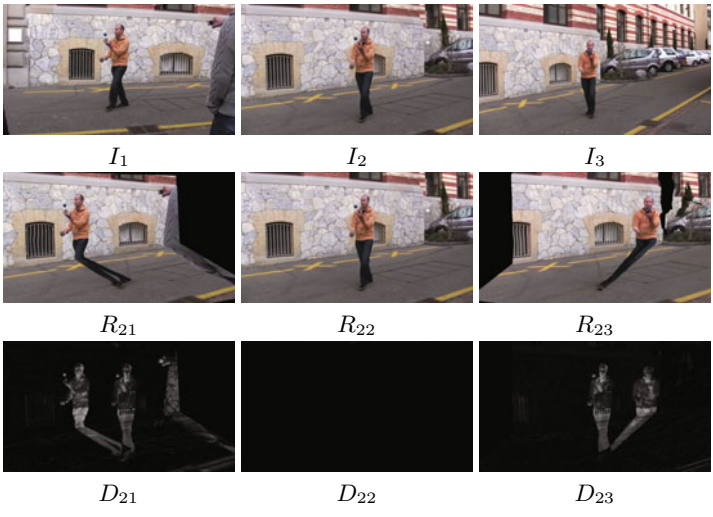


Fig. 2. (Top row) Source images acquired respectively by camera 1, 2 and 3. (Middle row) Images R_{ij} computed by projecting the previous images into camera 2 (black pixels indicate missing color information, i.e., $\alpha = 0$). (Bottom row) Difference images D_{ij}. (Best viewed in color).

In the end, the color of the pixel p in R_{ij} is defined as follows

$$R_{ij}^{p} = I_j(\pi_j(\pi_i^{-1}([p, Z_i^p]^T))) \tag{2}$$

Let us note that no color information can be retrieved for pixels of R_{ij} that map outside the field of view of camera j and also for those which have no depth information in Z_i, e.g., for those projecting onto regions not modeled by the background geometry. We keep track of such pixels by defining a binary mask α_{ij} such that, $\alpha_{ij}^{p} = 0$ indicates the absence of color information at pixel p in R_{ij}. The procedure of computing R_{ij} is performed in GPU using shaders.

Figure 2 shows some example images R_{ij} obtained by projecting the images captured by three different cameras, namely #1, #2 and #3, into the camera #2. The background geometry, in this case, models both the building and the street but it does not include the juggler. The reader can notice that, when the background geometry matches the current scene geometry the captured image I_i and the image R_{ij} look alike in all the pixels with α_{ij}^{p} equal to one. On the contrary, if the current scene geometry includes an additional object which was not present in the background geometry, this gets projected into the background points behind it. We refer to this reprojection as the *ghost* of the foreground object in the image R_{ij}. Figure 3 explains this concept visually. In Figure 2, the ghost of the juggler can be observed in both images R_{21} and R_{23} while it's not visible in R_{22} since the image is projected on itself.

By visually comparing R_{ij} and I_i, one can observe differences in the pixels belonging to foreground elements as well as in the pixels belonging to their

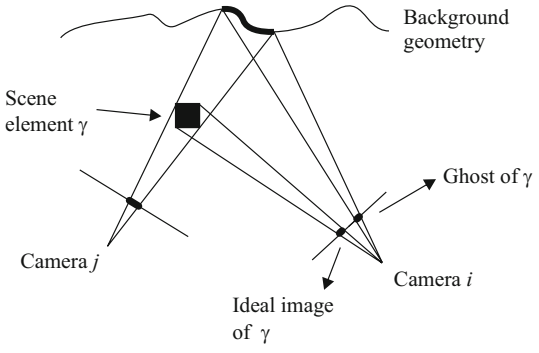


Fig. 3. Image formation process for a reprojection image R_{ij}. Since the scene element γ is not a part of the background geometry, it generates a ghost image on camera i which is far away from the region it should ideally project to if it were a part of the background geometry.

ghosts. Let's call D_{ij} the image obtained by a per-pixel comparison between image I_i and image R_{ij}. In order to make our comparison method robust to errors that may be present in either the calibration or in the background geometry, the similarity measure used to compare these two images takes into account for local affine transformations in the image space. We propose to compute D_{ij} as

$$D_{ij}^{p} = \min_{q \in W_p} \left(\left\| I_i^p - R_{ij}^q \right\| \right) \tag{3}$$

where W_p is a window around p and $\|\cdot\|$ is the L^1 norm in the RGB color space. This similarity measure proved to be more robust but, unfortunately, some details around the ghost borders are lost. This can be seen in Figure 4a where the ghost of the foreground object gets shrunk by half the window size used. In order to avoid these artifacts, the same approach is repeated by comparing, this time, the pixel p in R_{ij} to a corresponding window W_p in I_i. A result obtained by using this second approach is shown in Figure 4b where, this time instead, the silhouette of the foreground object gets shrunk by half the window size. In the end we chose to use the following metric which combines the advantages of the both the previous metrics:

$$D_{ij}^{p} = \max\left(\min_{q \in W_p} \left(\left\| I_i^p - R_{ij}^q \right\| \right), \min_{q \in W_p} \left(\left\| R_{ij}^p - I_i^q \right\| \right) \right) \tag{4}$$

A result obtained by applying this new metric can be seen in Figure 4c.

Given the input images I_i, all the possible images D_{ij} for each $i > j$ are computed. This leads to a set of $(n^2 - n)/2$ difference images D_{ij} that we will refer to as D. In the next paragraph the problem of recovering the 3D geometry of the foreground object is formulated in a probabilistic way using as observation the computed set of images D.

The scene to be reconstructed is discretized as a voxel grid. Let V be the random vector representing the occupancy state of all the voxels inside this grid

(a) (b) (c)

Fig. 4. Results obtained by applying different color similarity measures to compare the two images I_j and R_{ij} in order to build the image D_{ij}. (a) Result obtained by applying the Equation 3. (b) Result obtained by applying the Equation 3 with I_j and R_{ij} swapped. (c) Result obtained by applying the Equation 4.

where $V_k = 1$ indicates the voxel k is full and empty otherwise. The aim of our algorithm is to find a labeling L^* for V which maximizes the posterior probability $P(V = L|D)$, i.e.,

$$L^* = arg\,max_L P(V = L|D) \tag{5}$$

By the Bayes' rule, this is equivalent to

$$L^* = arg\,max_L(\log(P(D|V = L)) + \log(P(V = L))) \tag{6}$$

We first describe how the probability $P(D|V = L)$ is computed for a given labeling of the voxel grid, while $P(V = L)$ is described later.

Let ϕ_i^k denote the footprint of the voxel k in camera i, i.e., the projection of all the 3D points belonging to k onto the image plane of camera i. Furthermore, denote with χ_{ij}^k the set of the ghost pixels of voxel k in the image R_{ij}. Since these pixels are the ones corresponding to the background geometry points occluded by the foreground object in camera j, i.e. $\pi_j^{-1}([\phi_j^k, Z_j^{\phi_j^k}]^T)$, χ_{ij}^k can be computed as follows

$$\chi_{ij}^k = \pi_i(\pi_j^{-1}([\phi_j^k, Z_j^{\phi_j^k}]^T)) \tag{7}$$

i.e., by projecting those background points into camera i (See Figure 3).

We make three conditional independence assumptions for computing the probability $P(D|V = L)$: first, the state of the voxels are assumed to be conditionally independent; second, the image formation process is assumed to be independent for the all images and third, the color of a pixel in an image is independent from the others. Using these assumptions, the probability $P(D|V = L)$ can be expressed as

$$P(D|V = L) = \prod_k P(D|V_k = L_k) \tag{8}$$

where

$$P(D|V_k) = \prod_{i,j,p} P(D_{ij}^p|V_k) \quad \forall p \in \phi_i^k \cup \chi_{ij}^k \tag{9}$$

Let us now introduce another random variable C_{ij} representing the consensus between the pixels in image I_i and the ones in image R_{ij}. $C_{ij}^p = 1$ indicates that the color information at pixel p in I_i agrees with the color information at p in R_{ij}. Clearly, this variable strongly depends on the image D_{ij}.

Specifically, $P(D_{ij}^p|V_k)$ is modeled using a formulation similar to the one proposed by Franco and Boyer in [3], i.e.,

$$\begin{aligned} P(D_{ij}^p|V_k) = P(D_{ij}^p|C_{ij}^p = 1)P(C_{ij}^p = 1|V_k) + \\ P(D_{ij}^p|C_{ij}^p = 0)P(C_{ij}^p = 0|V_k) \end{aligned} \tag{10}$$

While in their work they used background images to determine $P(D_{ij}^p|C_{ij}^p)$ we assume the following: in case of consensus ($C_{ij}^p = 1$) the probability of D_{ij}^p being high is low and vice versa. Therefore $P(D_{ij}^p|C_{ij}^p = 1)$ is chosen to be a Gaussian distribution truncated for values smaller than 0. Concerning the pixels with no color information, i.e., the ones with $\alpha_{ij}^p = 0$, we assume this probability to be uniform and therefore,

$$P(D_{ij}^p|C_{ij}^p = 1) = \begin{cases} TG(D_{ij}^p) & \alpha_{ij}^p = 1 \\ U & \alpha_{ij}^p = 0 \end{cases} \tag{11}$$

where $TG(d)$ is the truncated Gaussian function and U the uniform distribution.

On the contrary, when there is no consensus ($C_{ij}^p = 0$) no information can be stated for D_{ij}^p and therefore $P(D_{ij}^p|C_{ij}^p = 0)$ is set to uniform distribution.

$P(C_{ij}^p = 1|V_k)$ and $P(C_{ij}^p = 0|V_k)$ are defined in a similar way as in [3] but while in their formulation, the state of the voxel k is influenced only by the background state of the pixels in ϕ_i^k, in our formulation its state is also influenced by the pixels in χ_{ij}^k. While this property adds additional dependence between the voxels, it provides more information on the state of each voxel. In fact, we not only rely on the consensus observed in the voxel's footprint ϕ_i^k but also on the consensus observed in χ_{ij}^k.

This allows us to recover from two kinds of situations, namely: when the colors of the foreground object are similar to the colors of the actual background points behind it, and when the information corresponding to the foreground object in the image R_{ij} is missing. However, our approach will not help if the colors of the actual background points in χ_{ij}^k are also similar to the colors of the foreground element.

Concerning $P(V = L)$ we assume dependency only between neighboring voxels. In this way, Equation 6 can be entirely solved using graph cuts [29,30,31]. More precisely, the pairwise potential $\log(P(V_a = L_a, V_b = L_b))$ between two neighboring voxels a and b is defined considering that if these voxels project to pixels lying on edges of the original images I_i there should be a low cost for cutting across these voxels and viceversa. To account for this, in our implementation, we compute the projection of the centers of each pair of neighboring voxels a and b on each image I_i. Subsequently we check all the pixels on the line connecting these two projections looking for an edge. If an edge is not found then the pairwise potential is increased.

To account for temporal continuity in the final mesh the voxel state prior takes into account its labeling computed in the previous frame according to $P(V_a = 1) = 0.3 + \xi(L_a^{*,t-1})$ where ξ defines the temporal smoothness. Once graph cuts provides a grid labeling L^* as a solution for Equation 6, marching cubes [32] is applied to obtain a continuous mesh of the dynamic object.

3 Results

The algorithm was tested on both real and synthetic data. For the real data test, we used a publicly available dataset provided by [16] where a juggler was filmed outdoors by six people holding cameras while some other people were walking by. Video streams have a resolution of 960×544 pixels at 25 fps. Background geometry was obtained using SfM+MVS on the available images of the dataset while the cameras were calibrated both photometrically and spatially using the techniques described in Section 2. About 300 frames of this sequence were processed by our method using a voxel grid of resolution $140 \times 140 \times 140$ covering the entire extent of the scene where the action took place.

Figures 1(a,d) show one reconstructed frame of this sequence where two persons are present in the scene. Figure 1c shows the reconstructed volume projected onto one of the cameras superimposed with the corresponding captured image. As the reader may notice, our system is also able to recover the shape of the balls being juggled by the performer. This however happens only in half of the reconstructed frames since motion blur is explicitly present in such parts of the image. Some more results are shown in Figures 5. Figures 5e and 5f show two situations where the algorithm does not work properly. In these two cases, the person walking behind is not visible in one of the views and is also occluded completely by the juggler in some of the other views. This leads to a noisy reconstruction.

This sequence was processed on a 2.93GHz Intel i7 computer with a NVIDIA GTX 285. For the chosen grid resolution, each frame took around 45 seconds to process. The current implementation however does not have any major optimizations, in fact only some parts of the code were implemented on GPU.

The results obtained for the juggler sequence were also compared with the ones obtained by applying standard background subtraction on the videos and then applying deterministic visual hull. The texture of the background geometry was estimated from the images used during the preprocessing stage. However, even small changes in the illumination or shadows in the scene did not allow us to infer accurate silhouettes for the performer. This is not an issue in our approach since only images taken at the same time instant are used for comparison.

The results were also compared with two state of the art techniques namely [20] and [16] but they were not convincing from a reconstruction point of view. In fact, [16] focuses on segmentation rather than reconstruction since that would be too sensitive to segmentation errors. A user interaction is also needed to label both foreground and background in some video frames. [20] assumes that the foreground is moving relative to the background, i.e., it is not moving rigidly with

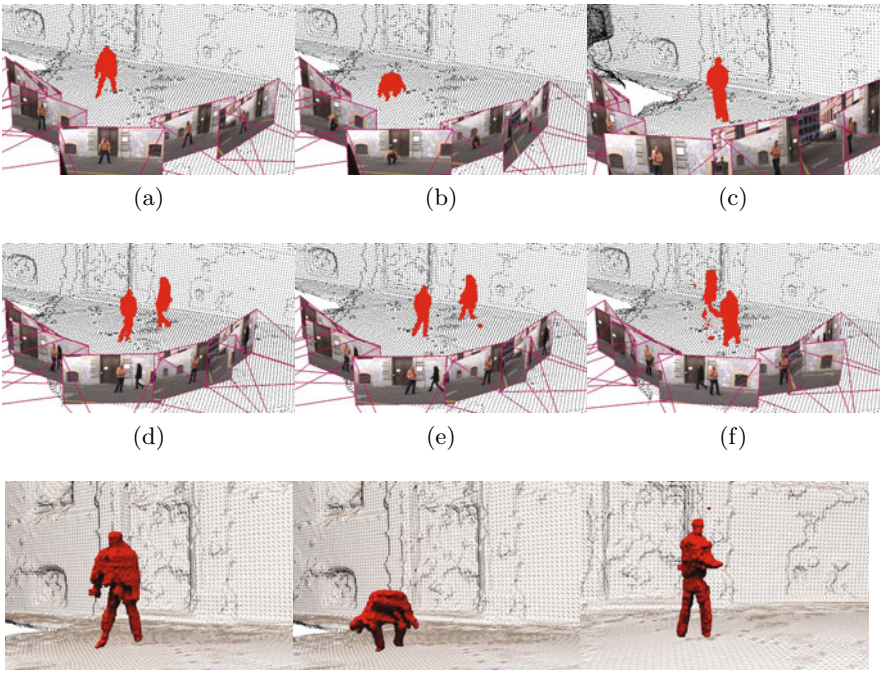


Fig. 5. (a-d) Results obtained using our approach. (e) and (f) show two cases where the algorithm does not behave properly due to strong occlusions between the two persons. (g,h,i) Reconstruction rendered from different viewpoints for the results in the top row. (Best viewed in color)

respect to the camera. However, this approach may fail in detecting objects or body parts moving slowly. This occurred frequently in the juggler sequence since, while the performer was juggling fast there was not much movement around his legs, and therefore they were often misclassified as background in the output obtained using [20]. Compared to the manually segmented silhouettes of the foreground objects [20] misclassified 25% of the pixels on an average while by projecting the volume computed with our method only 1% of the pixels were misclassified.

Some tests were performed on synthetic data to provide a quantitative evaluation of the algorithm. Using a commercial software, we rendered a scene with two balls bouncing in the center of a room filmed by 7 cameras moving in circle at a distance of 3m from the center of the action. The field of view of the cameras was 42° and the resolution of the video streams was 800×600. At first, the dataset reveals to be very simple and the algorithm performed an almost perfect reconstruction of the scene dynamics, obviously up to the chosen grid resolution. Therefore we rendered the dataset again introducing some ambiguities, more precisely, we textured the walls of the room with the same texture as

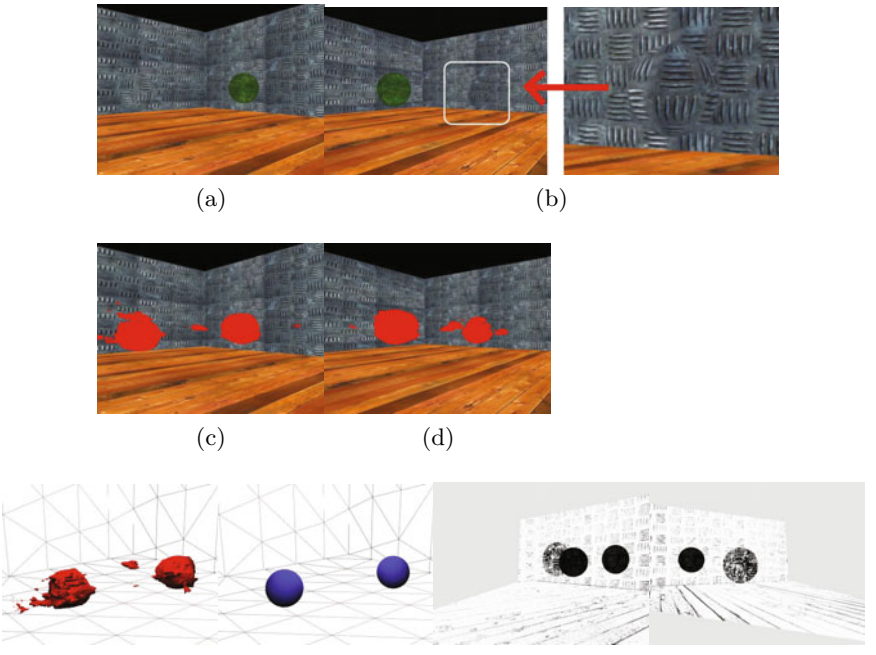


Fig. 6. (a,b) Two images from the synthetic sequence rendered from different cameras at the same time. The gray ball is barely distinguishable in these images. (c,d) Reconstructed volume projected back to the corresponding images in the above row. (e) Reconstructed volume. (f) Ground truth. (g,h) Two images D_{ij} computed during the shape estimation (the colors are inverted for visibility). (Best viewed in color)

one of the balls. Two frames of this new sequence are shown in Figures 6(a,b). As the reader can notice, even for a human it is difficult to visually distinguish between the gray ball and the gray wall.

A color based segmentation/visual hull technique will, in this case, either consider the entire wall as foreground or completely background, in both cases resulting in a bad reconstruction.

On the contrary, our reprojection based approach together with the robustness of the probabilistic framework was able to recover a reasonable reconstruction of the scene, as can be seen in Figure 6e. For a visual comparison, the ground truth is shown in Figure 6f.

The main reason why such a reconstruction can be achieved can be explained by looking at one of the D_{ij} images shown in Figures 6(g,h). While for a color based approach it is not feasible to distinguish between foreground and background in the case of the gray ball, if the texture of the background is provided by another camera some discrepancies between this texture and the observed image can be measured. A similar result can also be obtained if a per pixel color model of the background is available for each camera. However, since the

cameras are moving this model cannot be easily retrieved. Figures 6(c,d) show the reconstructed volume projected back to the respective original images.

We ran our algorithm on the full sequence consisting of 15 frames and the computed reconstructions were compared with the ground truth. At each frame, the error between the two models was evaluated numerically by measuring the average euclidean distance between the two surfaces. The average error for the whole sequence was 2cm, which corresponds to the used voxel size. The standard deviation for this error was 1.8cm. Note that, by definition the metric that we are using does not account for the sparse blobs in the reconstruction.

This error increases if we introduce inaccuracies in the background geometry and in the camera calibration. We ran the test again after adding Gaussian noise to the camera position with a standard deviation of 1.6cm and a uniformly distributed noise of $\pm$8cm to the background geometry. The average reconstruction error increased to 3.6cm, where the majority of the error was induced from the errors in calibration and not from errors in the geometry.

4 Conclusions

In this paper we proposed a new technique to model dynamic scenes in outdoor uncontrolled environments filmed with freely moving cameras. A probabilistic framework is proposed to deal with such a scenario and to provide a volumetric reconstruction of all the dynamic elements. The method exploits the structural information of the static parts of the scene, inferred in a preprocessing stage, to transfer the current background appearance across multiple cameras. Hence, it avoids the need to build a per-pixel background color model from temporal video data for segmentation, which is very challenging for scenes recorded with moving cameras.

Tests on synthetic data revealed a reconstruction accuracy of 2cm for footage filmed by 0.5MPixels cameras placed at a distance of 3m from the objects to be reconstructed. This error is relatively low considering the challenges present in the used dataset such as multiple occlusions and similar background/foreground colors (see Figure 6). Our approach reveals to be robust enough to deal with such ambiguities and also with calibration and geometry inaccuracies to an extent.

Experiments on real data proved the ability of our approach to recover the geometries of multiple dynamic objects filmed outdoors with freely moving cameras (see Figure 1). The reconstruction accuracy is not comparable with the one that other techniques can obtain for indoor controlled environments with static cameras. However, it must be noted that the scenario we used for our tests is much more challenging.

There are three main limitations of our approach. First, the algorithm depends on the possibility of estimating the color transfer function between the cameras which, in our case, was performed using a rather simple technique [28]. This works well in the tested sequences but, in the future, for a more generic scenario we should resort to a more complex calibration technique, like [33].

The second limitation is the resolution of the voxel grid which cannot be increased indefinitely without considering calibration and background geometry

errors. This limitation however, does not prevent us from recovering the small balls being juggled by the performer in half of the frames of the real data sequence.

The method inevitably inherits the limitations of the visual hull techniques on the class of reconstructible objects, i.e., it is not able to recover concave parts of the object if these concavities are not visible in at least one camera.

As a future extension, we plan to consider inside the proposed probabilistic framework other kinds of depth cues, like multiview stereo and narrow baseline stereo. A synergical fusion of these information will help overcome the last two limitations as well as increase the reconstruction accuracy.

Acknowledgements. We would like to thank Christopher Zach and David Gallup for their valuable help. The research leading to these results has received funding from the European Research Council under the European Community's Seventh Framework Programme (FP7/2007-2013)/ERC grant#210806.

References

1. Kim, H., Sarim, M., Takai, T., Guillemaut, J.Y., Hilton, A.: Dynamic 3d scene reconstruction in outdoor environments. In: 3DPVT (2010)
2. Guan, L., Franco, J.S., Pollefeys, M.: Multi-object shape estimation and tracking from silhouette cues. In: CVPR (2008)
3. Franco, J.S., Boyer, E.: Fusion of multi-view silhouette cues using a space occupancy grid. In: ICCV, pp. 1747–1753 (2005)
4. Furukawa, Y., Ponce, J.: Dense 3d motion capture for human faces. In: CVPR, pp. 1674–1681 (2009)
5. Tung, T., Nobuhara, S., Matsuyama, T.: Complete multi-view reconstruction of dynamic scenes from probabilistic fusion of narrow and wide baseline stereo. In: ICCV (2009)
6. Seitz, S., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: CVPR (2006)
7. Seitz, S.M., Dyer, C.R.: Photorealistic scene reconstruction by voxel coloring. In: CVPR, p. 1067 (1997)
8. Vlasic, D., Peers, P., Baran, I., Debevec, P., Popović, J., Rusinkiewicz, S., Matusik, W.: Dynamic shape capture using multi-view photometric stereo. In: SIGGRAPH Asia (2009)
9. Ahmed, N., Theobalt, C., Dobrev, P., Seidel, H.P., Thrun, S.: Robust fusion of dynamic shape and normal capture for high-quality reconstruction of time-varying geometry. In: CVPR (2008)
10. Vedula, S., Baker, S., Seitz, S., Kanade, T.: Shape and motion carving in 6d. In: CVPR (2000)
11. Matusik, W., Buehler, C., Raskar, R., Gortler, S.J., McMillan, L.: Image-based visual hulls. In: SIGGRAPH, pp. 369–374. ACM Press, New York (2000)
12. Goldlucke, B., Ihrke, I., Linz, C., Magnor, M.: Weighted minimal hypersurface reconstruction. PAMI, 1194–1208 (2007)
13. Hilton, A., Starck, J.: Multiple view reconstruction of people. In: 3DPVT (2004)
14. Sinha, S.N., Pollefeys, M.: Multi-view reconstruction using photo-consistency and exact silhouette constraints: A maximum-flow formulation. In: ICCV, pp. 349–356 (2005)

15. Hasler, N., Rosenhahn, B., Thormahlen, T., Wand, M., Gall, J., Seidel, H.P.: Markerless motion capture with unsynchronized moving cameras. In: CVPR (2009)
16. Ballan, L., Brostow, G.J., Puwein, J., Pollefeys, M.: Unstructured video-based rendering: Interactive exploration of casually captured videos. SIGGRAPH (2010)
17. Baumberg, A., Hogg, D.: An efficient method for contour tracking using active shape models. In: Motion of Non-Rigid and Articulated Objects, pp. 194–199 (1994)
18. Leibe, B., Cornelis, N., Cornelis, K., Gool, L.V.: Dynamic 3d scene analysis from a moving vehicle. In: CVPR (2007)
19. Elgammal, A., Duraiswami, R., Harwood, D., Davis, L.S.: Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. Proceedings of the IEEE 90, 1151–1163 (2002)
20. Sheikh, Y., Javed, O., Kanade, T.: Background subtraction for freely moving cameras. In: ICCV (2009)
21. Ivanov, Y., Bobick, A., Liu, J.: Fast lighting independent background subtraction. International Journal of Computer Vision 37, 199–207 (2000)
22. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004) ISBN: 0521540518
23. Pollefeys, M., Van Gool, L., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., Koch, R.: Visual modeling with a hand-held camera. IJCV 59, 207–232 (2004)
24. Zach, C., Pock, T., Bischof, H.: A globally optimal algorithm for robust TV-L^1 range image integration. In: ICCV (2007)
25. Zhang, Z.: A flexible new technique for camera calibration. PAMI 22 (2000)
26. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV 60, 91–110 (2004)
27. Haralick, R.M., Lee, C.N., Ottenberg, K., Nölle, M.: Review and analysis of solutions of the three point perspective pose estimation problem. IJCV 13 (1994)
28. Reinhard, E., Adhikhmin, M., Gooch, B., Shirley, P.: Color transfer between images. In: Computer Graphics and Applications, vol. 21, pp. 34–41. IEEE, Los Alamitos (2001)
29. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. PAMI 23, 1222–1239 (2001)
30. Kolmogorov, V., Zabin, R.: What energy functions can be minimized via graph cuts? PAMI 26, 147–159 (2004)
31. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. PAMI 26, 1124–1137 (2004)
32. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. SIGGRAPH 21, 163–169 (1987)
33. Kim, S., Frahm, J., Pollefeys, M.: Radiometric calibration with illumination change for outdoor scene analysis. In: CVPR, pp. 1–8 (2008)

Learning Image Structures for Optimizing Disparity Estimation

M.V. Rohith and Chandra Kambhamettu

Video/Image Modeling and Synthesis (VIMS) Lab, Department of Computer and Information Sciences, University of Delaware, Newark, DE, USA

Abstract. We present a method for optimizing the stereo matching process when it is applied to a series of images with similar depth structures. We observe that there are similar regions with homogeneous colors in many images and propose to use image characteristics to recognize them. We use patterns in the data dependent triangulations of images to learn characteristics of the scene. As our learning method is based on triangulations rather than segments, the method can be used for diverse types of scenes. A hypotheses of interpolation is generated for each type of structure and tested against the ground truth to retain only those which are valid. The information learned is used in finding the solution to the Markov random field associated with a new scene. We modify the graph cuts algorithm to include steps which impose learned disparity patterns on current scene. We show that our method reduces errors in the disparities and also decreases the number of pixels which have to be subjected to a complete cycle of graph cuts. We train and evaluate our algorithm on the Middlebury stereo dataset and quantitatively show that it produces better disparity than unmodified graph cuts.

1 Introduction

Stereo image based scene reconstruction has disparity calculation at its heart. There have been numerous attempts at solving the problem of disparity estimation. A taxonomy of such algorithms based on matching cost, aggregation and optimization can be found in [1]. A comparison based on errors in the calculated disparity may be found in [2] where authors classify algorithms into those suitable for view interpolation and others for structure reconstruction. Methods such as graph cuts and belief propagation generate dense disparity by formulating the problem as an optimal labeling problem characterized by a Markov Random field. The complexity of such dense estimation algorithms in memory and time increases with disparity range. There have also been methods aiming towards decreasing the memory and running times while still providing dense estimates. Some common techniques to simplify the problem are multiscale disparity calculation [3] and quad-tree decomposition [4]. As noted in [5], multiresolution and multiscale algorithms suffer with a problem in homogeneous regions. Most of the dynamic programming methods have their complexity decided predominantly by

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 627–640, 2011.

Fig. 1. Similar image structures in two scenes

the number of pixels and disparity range rather than the image characteristics (such as number of edges or number of corners). We pose a question, whether we can devise an algorithm that exploits the complexity (or simplicity) in the image and the similarity in structures common to images, in order to estimate disparity more effectively.

For example, in Figure 1, we see that there are similar structures in the two images of scenes characterized by two nearly parallel lines with no edges between them. There is similarity in the disparity maps of those sections as well. The disparity varies almost linearly in those regions indicating that the regions are planar. If the region is parallel to the baseline of the stereo setup, then the region will have constant disparity. In this case, a method such as graph cuts will estimate the disparity of the entire region in one α-expansion move, as all the pixels in the region will have the same label. On the other hand, if the plane is tilted with respect to the baseline, the disparity will vary linearly across the region. In this case however, graph cuts may take several moves to label the region accurately as it has to iterate over multiple labels. Depending on the resolution of the image and the disparity range, these iterations may consume enormously long time. Note that this complexity is intrinsic to the method of solving MRFs and not due to choice of a particular smoothness energy function. For example, in the method of α-expansions, only a single label is considered in each expansion irrespective of the smoothness constraint. Also, considering smoothness among pixels that are far from each other leads to a super-modular energy function, which cannot be solved by graph cuts.

It is surprising how a simple tilt in 3D increases the complexity of disparity estimation. It also opens up an opportunity to explore whether we can identify and train on the disparity maps of such structures to make the estimation in a future scene faster. For example, one such optimization would be to suggest that whenever we find two parallel image edges with no significant edges between them, we calculate the disparities only at the four corners and interpolate to fill the region in between. If we applied such an optimization, the case of tilted plane above would be less complex. But how do we abstract away this notion of structures? For example, should the edges always be straight or is it sufficient if they are parallel curves? What if the region in between had no edges but a smooth gradient in color, would the interpolation still be valid? Are there other structures in the image that we can similarly exploit? It would not only be tedious and time consuming to answer each of the above question manually but the answers would depend greatly on the nature of images. Hence we suggest that such structures be learned automatically from a given set of training images. But in

order for us to train, we need to abstract the notions of parallel edges and planar interpolation. In this paper, we present a method that attempts to accomplish this, and we present results to support it. We restrict ourselves to structures of uniform color and whose disparity may be estimated from bilinear interpolation from the extremes. Learning shape priors has improved performance in the areas of object detection [6] and model fitting [7], and we wish to explore its application in the area of disparity estimation.

Learning structures can be used to handle two practical problems that are otherwise hard to solve in stereo: subpixel disparity and large disparity range. Subpixel disparity is needed when a high resolution (in the depth direction) reconstruction is expected from the stereo analysis. Graph cuts does not solve it efficiently as it involves introduction of a new label for each subpixel disparity value needed, and this is the case with most label optimization algorithms. We show that, with our method of identifying structures, at least commonly occurring image structures can be assigned smooth disparity.

Handling large disparity ranges (usually occurs when images are shot with a wide baseline or if image sizes are large) directly is also impractical for labeling algorithms. Most common solution to this problem is the use of pyramid schemes where disparity is often computed on scaled versions of the images and then propagated to the next level in scale. It has been shown that such methods perform poorly in presence of image noise and homogeneous regions [5]. The method proposed in this paper can be applied to handle large disparity range images, as we do not have to go through every single disparity value, for every pixel, in the expansion moves.

At the outset, we would like to make it clear that we are not competing against methods such as segmentation based stereo or robust plane fitting methods. They succeed in estimating disparity in scenes which follow assumptions such as, 'segmentation edges correspond closely to disparity edges', etc. However, it is not easy to extend such methods to scenes which do not satisfy their assumptions. For outdoor scenes with little texture, finding segmentation parameters that gives a suitable segmentation is very hard and varies greatly with scene content. We show however that stereo processing of such scenes can nonetheless be optimized using a method sensitive to the structures in the scene. Also, segmentation is commonly carried out in a global framework whereas triangulation is fairly local operation. This makes our algorithm more suitable for a parallel implementation such as in a general purpose GPUs as compared to segmentation and hence makes the idea worth exploring.

We review some recent methods of learning in stereo in section 2 and give an overview of our approach in section 3. This is followed by details of the solution in section 4. Results and conclusions are in sections 5 and 6 respectively.

2 Previous Approaches

Learning algorithms have been used in stereo for purposes of regularization, parameter estimation [8], learning pixels susceptible to errors [9] and learning

optimal paths to take in a stereo process [10]. [8] attempts to optimize the weights of the energy function by iteratively estimating disparity and updating the weights under an expectation maximization scheme. It processes an image pair in isolation and no information is carried from one stereo pair to another. On the other hand, [10] tries to improve the results of SSD matching using edge information from segmentation. Local image information such as texture orientation are used as attributes for training. They use Metropolis-Hastings sampler to decide the optimal move at each stage, and use simulated annealing to obtain the final disparity. We are close to this method in that we apply the information learned from ground truth of training images in calculating the test case disparity. But in our case, the application of learned data is not just for labeling the pixels as occluded or foreground. We assign the disparities to the pixels directly. In this sense, we are not refining the random network that is to be solved, but actually optimizing the process of solving for the solution on a fixed network. [9] tries to model the long range relationship between pixels using structured SVM. The long range interaction is modeled by formulating the network as a Conditional Random Field(CRF) and places great emphasis on detecting occluded pixels accurately. In this paper, we are aiming at a disparity map that facilitates reconstruction and as such accurate disparity in non-occluded regions takes more preference over detection of occlusion.

It has been reported in [11,12] that triangulation schemes may be used to accelerate multiview reconstruction algorithms such as those proposed in [13]. [12] adaptively subdivides the scene as it estimates disparity. However, these methods depend on the existence of texture in most regions to facilitate triangulation. This problem is somewhat alleviated in [14] which identifies textureless areas using patterns in matching errors and interpolates disparity from the boundaries. However, none of the above methods use the monocular views to provide a prior for the subdivision.

Enforcing the constraint of planarity over a region of disparity leads to a super-modular energy function, which cannot be directly minimized by graph cut algorithms. However, such constraints are shown to produce better disparities [15]. Recently, much attention is being paid to developing methods which minimize super-modular energy terms [16]. A method that has recently been successfully applied to stereo [15] and optical flow [17] is to fuse multiple solutions. In [15], Quadratic Pseudo-boolean Optimization (QPBO) is used to fuse stereo solutions provided by various stereo algorithms, called proposals, into a single solution using graph cuts based scheme. The method can enforce a second order prior on the final solution instead of the first order prior in [18], thus removing the fronto-parallel assumption on the scene. But once the various proposals are fused for one image pair, there is little information that can be carried forward to fuse the proposals of another image pair. Also, the complexity of the method prohibits the application to large disparity range images.

As noted previously, methods for learning shape priors over object categories is proposed in [6]. In this work, annotated images containing similar class of objects are used to learn a shape prior. The edge maps of each image are aligned

to derive the mean shape and the variance. The alignment of a test image is calculated with TPS-RPM algorithm, which provides an alignment consistent with the alignment transformations already learned in a specific category. Note that learning structures for stereo is significantly different from this problem, as it involves optimizing for calculation of disparity, not just consistent segments. Also, there is no annotation available to decide which regions are to be considered for learning. There may be patches from various objects merged into single homogeneous region, and the scenes may contain image structures for which no consistent interpolation scheme is available. These factors are taken into consideration in our approach that is presented in the next section.

3 Overview of Our Approach

We model disparity estimation as an optimal labeling for energy minimization over a Markov Random Field (MRF) as formulated in [18]. Hence, we consider

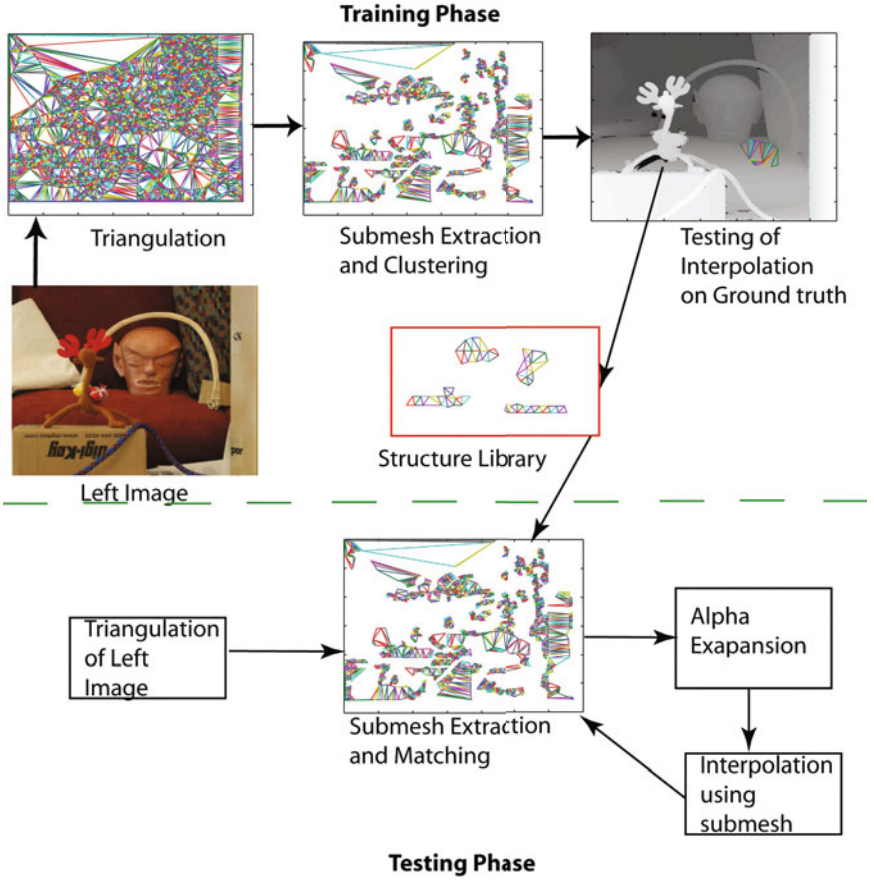


Fig. 2. Outline of our approach

only 4 neighbors for each pixel and use the same cost and smoothness functions as in [18]. Our goal is to minimize the area of the image over which pure α-expansion step is applied. Below we provide a brief overview of our training and testing methods (summarized in Figure 2).

For training, each input set contains two views of a scene (stereo pair) together with its ground truth disparity. The left image of the stereo pair undergoes a data dependent triangulation resulting in a set of triangles whose edges include the edges in the image. The triangle mesh is then processed to provide structural patterns which are a collection of neighboring triangles, which we call submeshes. A fixed number of neighbors (we choose 8 to 16) of a given triangle whose mean color and area are close to the given triangle are considered. Each submesh is characterized by the displacement vectors between the centroids as in Figure 3. The vectors are scaled with respect to the largest vector and rotated to align the largest vector with the horizontal to make them invariant to scale and rotation. The vectors are sorted by their magnitude and these form the attribute of the given triangle submesh. This process is applied randomly on the triangulation until all the triangles have been covered. The resulting attribute set is treated as a Gaussian mixture model and clusters are identified in this attribute space. Now we have to see if submeshes belonging to the same cluster have similar disparity. To judge the efficacy of a cluster for disparity estimation, we compute an approximate disparity by sampling the ground truth disparity at 4 extremes of the submesh and using bilinear interpolation to fill the pixels inside the submesh. The interpolated values are compared with those in ground truth disparity to obtain the error introduced. The clusters are ranked with respect to the errors and only those which have low error are retained. This process is applied to all the training image sequences. This results in a set of clusters each identified by a mean attribute and variance.

Given a test stereo pair, we triangulate the left image of the pair using similar technique as in training. For calculating the disparity, we alternate between the α-expansion step and the cluster interpolation step. We pick a triangle at random and check if the surrounding submesh is part of any cluster from training. If it is, then we run an α-expansion steps of all labels restricted to the 4 extreme triangles in the submesh. The disparity within the submesh is calculated by interpolation. This process is repeated till all the triangles are covered in a some submesh.

4 Details of Solution

4.1 Triangulation

In this paper, we will assume that scene is made up of piecewise planar patches. According to [19], patches of the scene having constant reflectance cannot have their geometry recovered by multiple views (unless all the rays which describe their silhouette are captured, which is not always possible in two views). Hence we may approximate textureless regions by planar regions and have disparity change linearly over them. This approximation is justified because presence of

large textureless region in the image indicates that the surface normal is varying smoothly across the surface that produced the image. Hence, the region has a large radius of curvature (small Gaussian curvature). The deviation of this surface from a plane is too small to be captured by the given camera pair under the given illumination. Thus, a plane seems to be the best approximation in such regions. As a corollary to the above, we may also argue that sufficiently rapid changes in the surface normal will produce an image discontinuity. Hence we have the following rules relating the image discontinuity to disparity discontinuity

1) Textureless area implies linear disparity change
2) Disparity discontinuity implies image edge

Note however that image edge does not imply disparity discontinuity as there may be surface features or shadows which cause image edges. We extend the definition of textureless areas above to regions containing no edge. (1) will not hold true in general after this extension, however results show that the approximation is valid for a variety of images. We have found that Canny edge detector provides good estimation of edge pixels for this purpose. If we assume that edges are almost straight, the disparity on the edge must vary linearly as well (because we assumed the scene to be made up of linear patches). So to describe the disparity of a patch, we do not need the disparity along all the edge points, rather we need it only at the corner points of the edge. Hence we define the initial features as Harris corner of the Canny edge image. But such corners may be distributed far from each other which leads to difficulty in defining their connectivity. So we sample the edges to produce points, which are farther from each other than a given threshold (we choose a threshold based on the image size). We will call such points as *essential points*. In describing the algorithms we need to make distinction between the edges contained in the image and the edges produced by the Delaunay triangulation of the essential points. We start with constraining Delaunay triangulation of the essential points, with the constraints being that points lying on the same edge in the image be connected by an edge in triangulation. The constraint is necessary to preserve the connectivity of essential points in regions containing multiple edges close to each other.

4.2 Structures and Clusters

Having obtained a triangulation, we need to identify the submeshes, set of triangles suitable for disparity interpolation, in the triangulation. We cannot take arbitrary neighborhoods of the triangle as submesh because they may belong to different objects. Hence we aim at extracting structures in which there is not much color variation. Also we assume that each triangle has at least one neighboring triangle belonging to the same object, that is - no object/scene component is within a single triangle. Given a triangle, we decide whether or not its neighbors are in the same color-submesh with the following steps:

1. Calculate the mean color of each neighbor and sort the neighbors in ascending order of difference in mean colors with respect to given triangle.

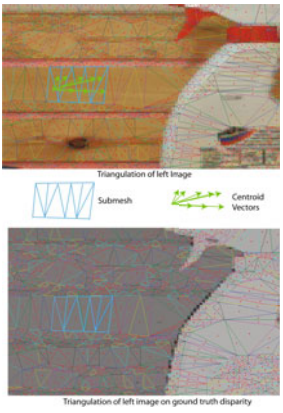


Fig. 3. Illustration of a single submesh and its centroid vectors

2. The neighbor with least error is always in the color-submesh.
3. If the error value of second neighbor is closest to that of third, then both of them are not in color-submesh.
4. Else, second neighbor is in the color-submesh.

The result of the above procedure is a list of triangles which are nearly homogeneous in color. We also constrain the triangles in the submesh to be similar with respect to their areas. The method described above for detecting color similarity is extended to that for area. A similar process which considers the area of the triangles instead of the mean color creates another triangle set called the area-submesh. The intersection between the color-submesh and the area-submesh is considered as the final submesh. It is ensured that the vertices of all the triangles in the submesh belong to a single connected graph, i.e, there are no isolated triangles. Such a parameter-less method ensures that the submesh identification does not depend on parameters such as image noise or sampling. The submesh is expanded until it includes a given number of triangles. Once all the submeshes are identified, the displacement vectors from the centroid of initial triangle to all others are calculated. These vectors are aligned so that the largest one points in the positive horizontal direction. The coordinates are then converted to a single column vector, and are stored as the attribute of the submesh. When attributes of all the submesh in a triangulation are obtained, the distribution of attributes is approximated by a Gaussian Mixture Model (GMM). We use the EM method for approximating the mixture. Given the distribution, we fit a model with a given number of components. This results in a posterior probability assignment for each point in the distribution as given by a several Gaussian distributions each characterized by a mean and a covariance matrix. We then cluster the points based on the probabilities and select those with sufficient support. Larger the number of triangles in a submesh, the more gain we will achieve when applied to stereo, however there will be fewer submeshes found. We experimented with several values and found 16 to be optimum submesh size for the images in

the evaluation dataset. The number of components that can be detected using GMM can be estimated by observing the number of variables to be estimated under the EM scheme, each new cluster needs estimation of a vector (mean) and a square symmetric matrix (covariance). Another criterion by which we choose the components is that the conditioning number of the covariance matrix should be above the working precision of the machine. This resulted in 10 to 15 clusters per image pair. For each cluster, the impact on using it for disparity interpolation is assessed. For each submesh in a cluster, the ground truth disparity is sampled at 4 extreme vertices (degenerate cases are excluded). The bilinear interpolated disparity from the 4 samples is compared with the ground truth in the submesh, and the mean error is noted. This is repeated for all meshes in all clusters. Only those clusters which have the low reprojection errors are retained (the threshold is calculated as mean reprojection error in training image pairs and ground truth disparity).

4.3 Disparity Calculation

Once the clusters are created from all training pairs, we can store the cluster means, variance and their mean error during training in a library. Given a new image pair, we start by triangulating the left image. Then a random triangle is chosen and a submesh created in a manner similar to training phase. It is

Algorithm 1. Summary of our method

Learning
1. For each training image, obtain essential points based on image edges and obtain triangulation as described in 4.1.
2. Identify submeshes in the triangulation and cluster them using the method in 4.2.
3. Filter to retain only those submeshes where bilinear interpolation fits the disparities of the pixels contained.
4. Construct Gaussian mixture model of the submeshes with N models using the features described in 4.2 to obtain a library of submeshes.
Calculating disparity for a new image
1. For each testing image, obtain essential points based on image edges and obtain triangulation as described in 4.1.
2. Identify submeshes in the triangulation and cluster them using the method in 4.2.
3. Check which of the submeshes can be filled up with interpolation by matching them with the models in the GMM (the feature vector should lie within one standard deviation of at least one of the means).
4. Run one α-expansion step to solve for disparity.
5. If there are any submeshes whose boundary points are assigned disparity, then interpolate within using bilinear interpolation.
6. Retain the disparity if reprojection error is less than threshold (estimated from mean reprojection error from training image pairs). Otherwise, mark the submesh as not interpolatable.
7. Repeat steps 4 through 6 till all the pixels are assigned disparity.

checked if the submesh belongs to any of the clusters in the library, the criterion for this being if the posterior probability as predicted by that cluster falls within standard deviation from the mean. If so, an α-expansion step is run on the extreme triangles in the submesh and the disparities within are filled using bilinear interpolation. There might be a few outliers in the filled-in region due to artifacts of sampling or edge detection. Hence, after filling in the disparities, any pixels whose re-projection is high is marked as unlabeled. This process is carried forward until all the triangles have been covered. Since there might be some triangles which do not belong to any cluster, we run an α-expansion on all such pixels at the end. The entire algorithm is summarized in Algorithm 1.

5 Results

The algorithm was implemented in Matlab, and the graph cuts implementation from [20] was used for α-expansions. The experiments were carried out on the 2005 and 2004 Middlebury stereo database pairs. The strategy of leave-one-testing was employed as in [9]. We chose constant weights for the graph cuts stage given by the automatic estimator in [20]. There were nearly 10,000 triangles in each triangulation, resulting in around 1000 submeshes in each image. There were nearly 20 clusters in each image corresponding to various strip like structures of various aspect ratios and also a small number of complex meshes. The clustering was done independently on each image and then the clusters were combined leaving out those from the current test image. The results for the Arts scene is shown in Figure 4 and the results of comparison with ground truth are seen in Table 1. Since we have not used any image segmentation to arrive at our result, or a sophisticated occlusion model, we restrict our comparison to learning method in [9] and [20]. Compared to the implementation provided by [20], we significantly reduced the number of α-expansions required in most scenes. This may be seen in Table 2. We also performed experiments to determine the contribution of the different aspects

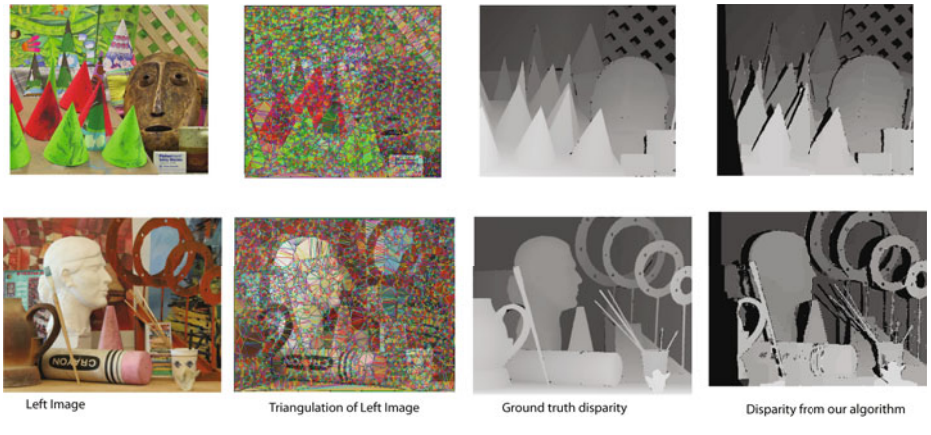


Fig. 4. Results on the Cone and Art scenes Middlebury 2005 Database

Table 1. Comparison of errors on ground truth from Middlebury dataset images. Error indicates percentage of bad pixels in non-occluded regions calculated as per [1]. Lower numbers are better. Training was based on leave-one-out strategy as in [9]. In our method, N denotes the number of triangles in a submesh.

Scene	Art	Books	Laundry	Reindeer	Teddy	Cones	Moebius
[20], Graph cuts	-	-	-	-	11.2	5.36	-
[9], Grid (K=1)	14.66	19.12	19.16	11.72	11.34	4.68	10.88
Our Method (N=8)	13.34	19.23	17.12	11.66	10.32	4.61	9.43
Our Method (N=16)	12.96	18.80	16.54	11.12	8.17	5.11	8.96

Table 2. Above table shows the effectiveness of using structure in each image pair. The entries are percentage of pixels in the image. *Not matched* represents pixels which could not be placed in any known submesh. *Interpolated* indicates those which were successfully labeled because of submesh detection. *Interpolation failed* represents the pixels that were predicted wrongly by interpolation. In an unmodified graph cuts, 100% of the pixels would be procesed using graph cuts. By using our scheme only the first and the last portion of pixels need to be processed using graph cuts. The table shows 3 schemes for detecting regions which can be interpolated. First method uses color alone, second uses size alone and the third uses both.

Method	Pixels	Art	Books	Laundry	Reindeer	Teddy	Cones	Moebius
	Not Matched	52.31	43.59	64.05	27.10	47.81	61.25	69.67
N=8	Interpolated	13.4	12.12	14.66	9.92	12.43	15.43	5.35
(Color only)	Interpolation failed	34.29	44.29	21.29	62.98	39.76	23.32	24.98
	Not Matched	44.22	28.02	29.21	56.49	62.26	53.56	73.98
N=8	Interpolated	32.76	53.75	56.00	24.59	22.71	28.15	5.14
(Size only)	Interpolation failed	23.02	18.23	14.79	18.92	15.03	18.29	20.88
	Not Matched	32.11	41.23	12.01	45.02	30.11	42.73	64.03
N=8	Interpolated	55.68	50.84	79.20	45.09	62.61	50.81	22.01
(Both)	Interpolation failed	12.21	7.93	8.79	9.89	7.28	6.46	13.96

of the triangles towards the detection of regions that can be interpolated. As seen in Table 2, using color alone matches a large number of submeshes, however much of these are falsely classified and hence the interpolation fails for large number of pixels. Using size alone restricts the number of submeshes matched, however there is no significant increase in the number of good interpolations. Using both color and size however seems to produce a good detection strategy to identify pixels amenable to be assigned disparity by interpolation. Though the number of submeshes are fewer compared to those found by color alone, the number of good interpolations are found to increase.

Also since we have the grid network for the MRF solution, we compare with results from similar formulation in [9]. Figure 5 shows an example structure found in different images. We can observe that detection of the structure is invariant

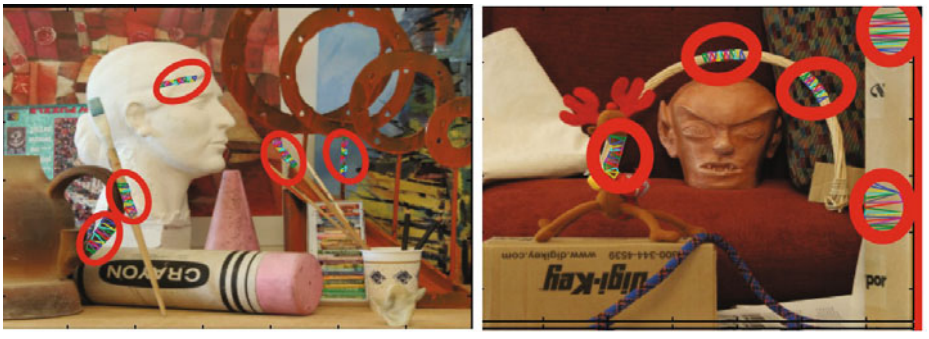


Fig. 5. Common structures found in the 3 scenes. This structure seems to correspond to long thin strip. Note that the structure has been identified over various ranges and orientation.

to orientation and scale changes. Also, the structures learned are generic enough to be reused across scenes.

6 Conclusion

We started with question of whether we can devise an algorithm that exploits the complexity (or simplicity) in the image and the similarity in structures common to images to estimate disparity more effectively. Enforcing long range interactions introduces super-modular energy terms in MRF formulation. Such energy functions cannot be easily minimized using methods such as graph cuts. Learning repetitive structures with similar disparity characteristics may help in complementing graph cuts. Our objectives were to model such structures with a common representation, cluster them according to their usefulness of disparity calculation, detect and apply them in stereo processing of other images with same characteristics. In this paper, we give one method of achieving the above for structures of uniform color and linearly varying disparity. We used a triangle based representation of the image, and identified submeshes as structures. Centroid displacements were used as characteristics of submeshes and proximity to ground truth was used as measure of usefulness in stereo. We designed a method that is able to extract simple structures effectively and lead to better disparity estimation process. Towards this end, we demonstrated a learning algorithm which identifies structures in scenes, and stores them in a library to calculate stereo of a new scene better. Our results show that the disparity estimated using our method is quantitatively better than those obtained from unmodified graph cuts. The number of α-expansion moves required is also shown to be decreased. Further work will involve using a two stage disparity estimation process which would support structures of varying color but similar depth.

Acknowledgements. This work was made possible by National Science Foundation (NSF) Office of Polar Program grant ANT0636726.

References

1. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal of Computer Vision 47, 7–42 (2002)
2. Kostková, J., Čech, J., Šára, R.: Dense stereomatching algorithm performance for view prediction and structure reconstruction. In: Bigun, J., Gustavsson, T. (eds.) SCIA 2003. LNCS, vol. 2749, pp. 101–107. Springer, Heidelberg (2003)
3. Felzenszwalb, P., Huttenlocher, D.: Efficient belief propagation for early vision. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004, vol. 1, pp. I–261–I–268 (2004)
4. Leung, C., Appleton, B., Sun, C.: Iterated dynamic programming and quadtree subregioning for fast stereo matching. Image Vision Comput. 26, 1371–1383 (2008)
5. Trinh, H.: Efficient stereo algorithm using multiscale belief propagation on segmented images (2008)
6. Tingting Jiang, F.J., Schmid, C.: Learning shape prior models for object matching. In: Proc. Computer Vision and Pattern Recognition Conf. (2009)
7. Besbes, A., Nikos Komodakis, G.L., Paragios, N.: Shape priors and discrete mrfs for knowledge-based segmentation. In: Proc. Computer Vision and Pattern Recognition Conf. (2009)
8. Zhang, L., Seitz, S.M.: Parameter estimation for mrf stereo. In: CVPR 2005: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), Washington, DC, USA, pp. 288–295. IEEE Computer Society, Los Alamitos (2005)
9. Li, Y., Huttenlocher, D.P.: Learning for stereo vision using the structured support vector machine. In: CVPR. IEEE Computer Society, Los Alamitos (2008)
10. Kong, D., Tao, H.: A method for learning matching errors for stereo computation (2004)
11. Labatut, P., Pons, J.P., Keriven, R.: Efficient multi-view reconstruction of large-scale scenes using interest points, delaunay triangulation and graph cuts. In: IEEE International Conference on Computer Vision, pp. 1–8 (2007)
12. Wey, P., Fischer, B., Bay, H., Buhmann, J.M.: Dense stereo by triangular meshing and cross validation. In: Franke, K., Müller, K.-R., Nickolay, B., Schäfer, R. (eds.) DAGM 2006. LNCS, vol. 4174, pp. 708–717. Springer, Heidelberg (2006)
13. Strecha, C., Fransens, R., Gool, L.V.: Wide-baseline stereo from multiple views: A probabilistic account. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 552–559 (2004)
14. Rohith, M., Somanath, G., Kambhamettu, C., Geiger, C.: Towards estimation of dense disparities from stereo images containing large textureless regions. In: 19th International Conference on Pattern Recognition, ICPR 2008, pp. 1–5 (2008)
15. Woodford, O.J., Torr, P.H.S., Reid, I.D., Fitzgibbon, A.W.: Global stereo reconstruction under second order smoothness priors. In: CVPR. IEEE Computer Society, Los Alamitos (2008)
16. Komodakis, N., Paragios, N.: Beyond pairwise energies: Efficient optimization for higher-order mrfs. In: Proc. Computer Vision and Pattern Recognition Conf. (2009)
17. Lempitsky, V.S., Roth, S., Rother, C.: Fusionflow: Discrete-continuous optimization for optical flow estimation. In: CVPR. IEEE Computer Society, Los Alamitos (2008)

18. Kolmogorov, V., Zabih, R.: Computing visual correspondence with occlusions using graph cuts. In: Proceedings of Eighth IEEE International Conference on Computer Vision, ICCV 2001, vol. 2, pp. 508–515 (2001)
19. Baker, S., Sim, T., Kanade, T.: A characterization of inherent stereo ambiguities. In: Proceedings of the 8th International Conference on Computer Vision, pp. 428–435. IEEE Computer Society Press, Los Alamitos (2001)
20. Kolmogorov, V., Zabih, R.: Multi-camera scene reconstruction via graph cuts. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2352, pp. 82–96. Springer, Heidelberg (2002)

Image Reconstruction for High-Sensitivity Imaging by Using Combined Long/Short Exposure Type Single-Chip Image Sensor

Sanzo Ugawa, Takeo Azuma, Taro Imagawa, and Yusuke Okada

Advanced Tecnology Reserch Laboratories, Panasonic Corporation,
3-4 Hikaridai, Seika-cho, Sagara-gun, Kyoto 619-0237, Japan
ugawa.sanzo@jp.panasonic.com

Abstract. We propose a image reconstruction method and a sensor for high-sensitivity imaging using long-term exposed green pixels over several frames. As a result of extending the exposure time of green pixels, motion blur increases. We use motion information detected from high-frame-rate red and blue pixels to remove the motion blur. To implement this method, both long- and short-term exposed pixels are arranged in a checkerboard pattern on a single-chip image sensor. Using the proposed method, we improved fourfold the sensitivity of the green pixels without any motion blur.

1 Introduction

Recently, in the field of video input, the number of pixels has been continuously increasing while the pixel pitch has been decreasing. As pixels continue to shrink, the amount of light that can be received within a certain exposure time is reduced. Consequently, securing a higher Signal-to-Noise-Ratio (SNR) is becoming difficult.

To improve the SNR in the field of image sensing, technologies such as on-chip microlens [15] and back side illumination [12, 19] have been proposed. With the former, an on-chip microlens is structured on each pixel to focus incident light on a photodiode. This method triples the sensitivity. With the latter, the image sensor of back side illumination is wired under the transistor. The development of a sensor with a pixel size of 1.4 um that provides twice the sensitivity of conventional sensors has been reported [17, 21]. Also, the use of a sensor with a white pixel in the color filter area was proposed [11, 14].

For high resolution imaging using a small sensor, long-term exposure can increase the amount of incident light. However, motion blur occurs if the object of the image moves. The use of a combination approach with multiple images to reduce space-invariant blur has also been proposed [5, 6, 16, 18, 22]. Recently, a coded sampling method was proposed [3]. However, these methods cannot be used in small cameras because they require a number of image sensors.

We propose a image reconstruction algorithm and its dedicated sensor shown in Figure 1. The sensing method can effectively input a lot of light and motion

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 641–652, 2011.

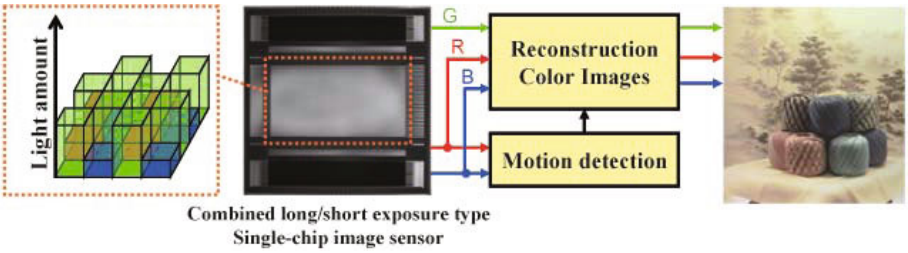


Fig. 1. With the proposed method images of G are obtained by long-term exposure. Motion blur of G is reduced by the pixel values of R and B in image processing.

information by mixing both long- and short-term exposed pixels on a single sensor. Additionally, we arranged the RGB on the basis of both human vision system and color correlation along the spectra of natural scenes.

The rest of the paper is as follows: We propose combined long/short exposure type image sensing in Section 2 and the reconstruction algorithm in Section 3. The results of experiments are shown in Section 4. We discuss the results in Section 5 and conclude the paper in Section 6.

2 Use of Single-Chip Image Sensor for Combined Long/Short Exposure Type Image Sensing

To improve the sensitivity and high-resolution imaging of a single-chip image sensor, our sensing method uses a long exposure time for some of the pixels on a single-chip image sensor. In this chapter, we explain the arrangement of both long- and short-term exposed pixels and the arrangement of RGB color filters in a single-chip image sensor.

2.1 Arrangement of Long-Term and Short-Term Exposed Pixels

In high-resolution imaging with a small image sensor, a long-term exposure is necessary to get the required amount of light. There is a trade-off relationship between the intervals of the exposure time and the frame-rate. Therefore, a long-term exposure decreases the frame-rate, and the captured image often includes motion blur if the object moves. However, this motion blur can be removed with motion information, so we then simultaneously take high-frame-rate images to obtain motion information.

In the image sensor, the difference between long- and short-term exposed images is the frame-rate. We define pixels which have a low-frame-rate image exposed over a long-term as 'long-term exposed pixels' and pixels which have a high-frame-rate image exposed over a short-term as 'short-term exposed pixels'.

We arranged a combination of both long-term exposed pixels and short-term exposed pixels in a single sensor. We also considered how to arrange each pixel for monochrome imaging. To efficiently estimate the motion information of

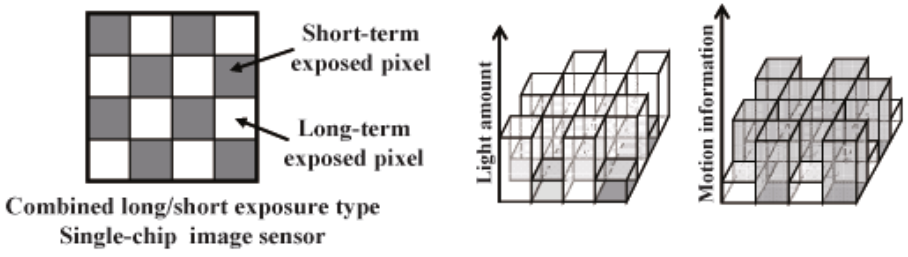


Fig. 2. Long- and short-term exposed pixels as checkerboard pattern in single-chip image sensor

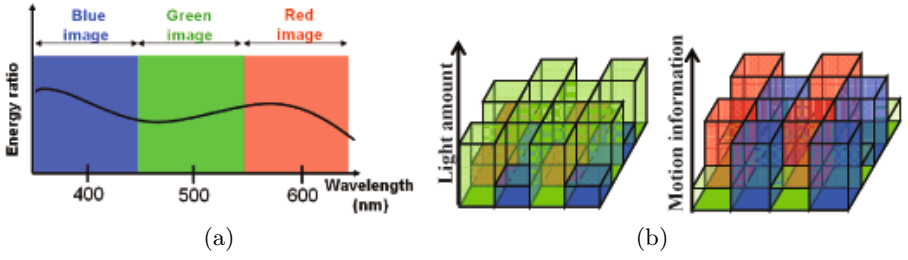


Fig. 3. (a) In the light spectrum, G is adjacent to B and R. (b) We arrayed G to long-term exposed pixels and R and B to short-term exposed ones.

long-term exposed pixels, the arrangement of short-term exposed pixels is homogeneously distributed. As shown in Figure 2, we arranged the pixels in a checkerboard pattern.

2.2 Arrangement of RGB Color Filters

We arranged RGB color filters in the sensor to reconstruct color images.

In the human vision system, spatial resolution for luminance is higher than that for hue, so we assigned green, which is the main factor in luminance, to the high-resolution image.

As shown in Figure 3(a), in the power spectrum of natural scenes, the power distribution of RGB are broad. The color of the adjoining wavelength has a high correlation. Both G-R and G-B are highly correlated. Hence, motion blur of G can be removed by using motion information calculated from R and B.

Based on both the properties of the human vision system and the color correlation among the spectra of natural scenes, we assigned G for long-term exposures and both R and B for short-term exposures, as illustrated in Figure 3(b). The motion information is detected from the R and B images because they have a high-frame-rate. This color arrangement is the same as that of the Bayer color arrangement [4].

3 Reconstruction of Color Images

We reconstructed color images from the sensing images explained in Section 2. The color images are calculated by minimizing a cost function. The cost function is composed of a difference between the ideal image and the sensing image with some regularization terms. In this section, we explain the cost function and how it can be solved by minimizing it.

3.1 Cost Function

Our method reconstructs color images by making use of the cost function. We define the ideal solution of RGB images as $\mathbf{f}$ and RGB sensed images as $\mathbf{g}_{\mathrm{R}}$, $\mathbf{g}_{\mathrm{G}}$, and $\mathbf{g}_{\mathrm{B}}$. $\mathbf{f}$ is a three-dimensional vector of R, G, and B and includes all RGB pixel values. Additionally, $\mathbf{f}$ includes R, G, and B values for each pixel position. The frame-rate of $\mathbf{f}$ is the same as the R/B, and the resolution of $\mathbf{f}$ is the same as the total number of pixels in a single sensor. The relation between $\mathbf{f}$ and the sensing RGB images, $\mathbf{g}_{\mathrm{R}}$, $\mathbf{g}_{\mathrm{G}}$, and $\mathbf{g}_{\mathrm{B}}$ is shown in Equation (1).

$$\begin{aligned} \mathbf{g}_{\mathrm{R}} &= \mathbf{H}_{\mathrm{R}}\mathbf{f} \\ \mathbf{g}_{\mathrm{G}} &= \mathbf{H}_{\mathrm{L}}\mathbf{H}_{\mathrm{G}}\mathbf{f} \\ \mathbf{g}_{\mathrm{B}} &= \mathbf{H}_{\mathrm{B}}\mathbf{f} \end{aligned} \tag{1}$$

As shown in Figure 2, $\mathbf{H}_{\mathrm{G}}$, $\mathbf{H}_{\mathrm{R}}$, and $\mathbf{H}_{\mathrm{B}}$ are degradation operators of G, R, and B pixel values in a Bayer pattern from the full pixel values, and $\mathbf{H}_{\mathrm{L}}$ is the degradation in frame-rate, which indicates long-term exposure. Equation (1) is a linear operation.

There are many candidates of $\mathbf{f}$ values that satisfy Equation (1). We add two regularization terms, $\mathbf{Q}_{\mathrm{S}}\mathbf{f}$ and $\mathbf{Q}_{\mathrm{m}}\mathbf{f}$, to specify $\mathbf{f}$. The terms $\mathbf{Q}_{\mathrm{S}}\mathbf{f}$ and $\mathbf{Q}_{\mathrm{m}}\mathbf{f}$ are based on the features of objects. The spatial smoothness can be given by $\mathbf{Q}_{\mathrm{S}}\mathbf{f}$, and the consistency of pixel value by the motion of the object can be given by $\mathbf{Q}_{\mathrm{m}}\mathbf{f}$. The cost function J that demands $\mathbf{f}$ is shown in Equation (2).

$$\begin{aligned} J = &||\mathbf{H}_{\mathrm{L}}\mathbf{H}_{\mathrm{G}}\mathbf{f} - \mathbf{g}_{\mathrm{G}}||^2 + ||\mathbf{H}_{\mathrm{R}}\mathbf{f} - \mathbf{g}_{\mathrm{R}}||^2 + ||\mathbf{H}_{\mathrm{B}}\mathbf{f} - \mathbf{g}_{\mathrm{B}}||^2 \\ &+ \lambda_{\mathrm{S}}||\mathbf{Q}_{\mathrm{S}}\mathbf{H}_{\mathrm{C}}\mathbf{f}||^2 + \lambda_{\mathrm{m}}||\mathbf{Q}_{\mathrm{m}}\mathbf{f}||^2 \end{aligned} \tag{2}$$

where λ_{S} and λ_{m} are regularization parameters. A detailed explanation of $\mathbf{Q}_{\mathrm{S}}\mathbf{f}$ and $\mathbf{Q}_{\mathrm{m}}\mathbf{f}$ is shown in the following subsection.

Color constraint term. $\mathbf{g}_{\mathrm{G}}$ is high resolution. Meanwhile, $\mathbf{g}_{\mathrm{R}}$ and $\mathbf{g}_{\mathrm{B}}$ are low resolution. When $\mathbf{g}_{\mathrm{G}}$ is spatially correlated with $\mathbf{g}_{\mathrm{R}}$ and $\mathbf{g}_{\mathrm{B}}$, the resolution of $\mathbf{g}_{\mathrm{R}}$ and $\mathbf{g}_{\mathrm{B}}$ can be improved by means of the correlation. On this point, we assume inter-RGB correlations to combine multiple color-channel images and implement them into the cost function as spatial color-smoothness. In natural images, the spatial smoothness differs depending on the basis of color space, we choose principal component vectors to define smoothness in order to set smoothness parameters effectively. The fourth and fifth terms in Equation (2)

are regularization terms that represent the spatial smoothness of the principal color of a reconstructed image sequence. First, the basis of the principal color space is defined by the principal component analysis of the RGB pixel values of natural images. Next, the principal color images are obtained by transformation matrices as follows:

$$\mathbf{f}_{\mathrm{C}i} = \mathbf{H}_{\mathrm{C}i}\mathbf{f}, i \in \{1, 2, 3\} \tag{3}$$

where $\mathbf{f}_{\mathrm{C}1}$, $\mathbf{f}_{\mathrm{C}2}$, and $\mathbf{f}_{\mathrm{C}3}$ represent the first, second, and third PCA color component images. The spatial smoothness of the pixel value in the principal-color image sequence is obtained by using $\mathbf{Q}_{\mathrm{S}}$ as a Laplacian operator. Here, we suppose that the pixel value varies smoothly in many regions in the reconstructed image, and the sum of squares of the second differentiation centered on each pixel is shown in Equation (4).

$$||\mathbf{Q}_{\mathrm{S}}\mathbf{H}_{\mathrm{C}}\mathbf{f}||^2 = \sum_{x=1}^{H}\sum_{y=1}^{W}\left\{\frac{\partial^2}{\partial x^2}\mathbf{f}_{\mathrm{C}i}(x,y,z) + \frac{\partial^2}{\partial y^2}\mathbf{f}_{\mathrm{C}i}(x,y,z)\right\}, i \in 1, 2, 3 \tag{4}$$

where H is the height of the image and W is the width of the image. The difference formulation of Equation (4) is shown in Equation (5).

$$\begin{aligned}||\mathbf{Q}_{\mathrm{S}}\mathbf{H}_{\mathrm{C}}\mathbf{f}||^2 = \sum_{x=1}^{H}\sum_{y=1}^{W}\Big\{&4\mathbf{f}_{\mathrm{C}i}(x,y,t) - \mathbf{f}_{\mathrm{C}i}(x-1,y,t) - \mathbf{f}_{\mathrm{C}i}(x,y-1,t)\\ &-\mathbf{f}_{\mathrm{C}i}(x+1,y,t) - \mathbf{f}_{\mathrm{C}i}(x,y+1,t)\Big\}, i \in 1, 2, 3\end{aligned} \tag{5}$$

Motion consistency term. We assume when an object moves, the pixel value does not change from the start to the end position. Motion information is detected by a sequence of three R/B color images, which are respectively captured at time t-1, t, and t+1. The relation of the time t-1 image and the time t image shows the motion toward the past, which is defined as (u_P,v_P). The relation of the time t image and the time t+1 image shows the motion toward the future, which is defined as (u_F,v_F). Motion information (u_P,v_P) and (u_F,v_F) are calculated as shown in Equation (6).

$$\begin{aligned}&\min_{u_P,v_P}\sum_{i=1}^{S_H}\sum_{j=1}^{S_W}|\mathbf{f}(x+i-u, y+j-v, t-1) - \mathbf{f}(x+i, y+j, t)|\\ &\min_{u_F,v_F}\sum_{i=1}^{S_H}\sum_{j=1}^{S_W}|\mathbf{f}(x+i-u, y+j-v, t+1) - \mathbf{f}(x+i, y+j, t)|\end{aligned} \tag{6}$$

S_H and S_W indicate a window size for motion detection. Pixel locations that correspond to G pixels in a Bayer pattern array are linearly interpolated by the surrounding R and B pixels to detect motion information at such pixels in advance. The regularization term regarding motion information is shown in Equation (7).

$$||Q_m f||^2 = \sum_{x=1}^{H}\sum_{y=1}^{W}\Big\{\mathbf{f}(x,y,t) - \mathbf{f}(x+u_P, y+v_P, t-1)\Big\}^2 + \sum_{x=1}^{H}\sum_{y=1}^{W}\Big\{\mathbf{f}(x,y,t) - \mathbf{f}(x+u_F, y+v_F, t+1)\Big\}^2 \tag{7}$$

We estimate sub-pixel motion by calculating the equiangular line fitting with the sum of absolute differences.

3.2 Minimization of the Cost Function

The output image $\mathbf{f}$ is obtained by minimizing the cost function J. Since J is the second form of $\mathbf{f}$, minimization of J always yields the global minimum of J. Thus, we can obtain $\mathbf{f}$ by differentiating J with respect to $\mathbf{f}$ and equating it with 0, as shown in Equation (8).

$$\begin{aligned}\frac{\partial J}{\partial \mathbf{f}} &= (\mathbf{H}_\mathrm{L}\mathbf{H}_\mathrm{G})^\mathrm{T}(\mathbf{H}_\mathrm{L}\mathbf{H}_\mathrm{G}\mathbf{f} - \mathbf{g}_\mathrm{G}) + \mathbf{H}_\mathrm{R}^\mathrm{T}(\mathbf{H}_\mathrm{R}\mathbf{f} - \mathbf{g}_\mathrm{R}) + \mathbf{H}_\mathrm{B}^\mathrm{T}(\mathbf{H}_\mathrm{B}\mathbf{f} - \mathbf{g}_\mathrm{B}) \\ &\quad + \sum_{i=1}^{3}\lambda_{\mathrm{C}i}(\mathbf{Q}_\mathrm{S}\mathbf{H}_{\mathrm{C}i})^\mathrm{T}\mathbf{Q}_\mathrm{S}\mathbf{H}_{\mathrm{C}i}\mathbf{f} + \lambda_\mathrm{m}\mathbf{Q}_\mathrm{m}^\mathrm{T}\mathbf{Q}_\mathrm{m}\mathbf{f} \\ &= 0\end{aligned} \tag{8}$$

Thus,

$$\begin{aligned}&\Big\{(\mathbf{H}_\mathrm{L}\mathbf{H}_\mathrm{G})^\mathrm{T}(\mathbf{H}_\mathrm{L}\mathbf{H}_\mathrm{G}) + \mathbf{H}_\mathrm{R}^\mathrm{T}\mathbf{H}_\mathrm{R} + \mathbf{H}_\mathrm{B}^\mathrm{T}\mathbf{H}_\mathrm{B} \\ &\quad + \sum_{i=1}^{3}\lambda_{\mathrm{C}i}(\mathbf{Q}_\mathrm{S}\mathbf{H}_{\mathrm{C}i})^\mathrm{T}\mathbf{Q}_\mathrm{S}\mathbf{H}_{\mathrm{C}i} + \lambda_\mathrm{m}\mathbf{Q}_\mathrm{m}^\mathrm{T}\mathbf{Q}_\mathrm{m}\Big\}\mathbf{f} \\ &\qquad = \Big\{(\mathbf{H}_\mathrm{L}\mathbf{H}_\mathrm{G})^\mathrm{T}\mathbf{g}_\mathrm{G} + \mathbf{H}_\mathrm{R}^\mathrm{T}\mathbf{g}_\mathrm{R} + \mathbf{H}_\mathrm{B}^\mathrm{T}\mathbf{g}_\mathrm{B}\Big\}\end{aligned} \tag{9}$$

Equation (9) can be solved by the conjugate gradient method.

4 Experimental Results

In this Section, we describe our simulation experiments. Additionally, we show how the developed image sensor is used and demonstrate that our method is effective.

4.1 Simulation

Our method was found to be effective even under low illumination. We used Peak-Signal-to-Noise-Ratio (PSNR) for evaluation. PSNR is calculated as follows:

$$PSNR(\mathbf{f}_{calc}) = 20\log_{10}\frac{S}{\sqrt{\frac{1}{N}||\mathbf{f}_{true}(x,y,t) - \mathbf{f}_{calc}(x,y,t)||^2}} \tag{10}$$

Table 1. Experimental conditions

Item	Parametaer
Number of scenes	20
Number of frames for each scene	180 frames
Interval of long-term exposure	4 frames
Noise level	8.06 (PSNR=30.0dB)

Table 2. Results of PSNR for Bayer reconstruction and proposed method

	Scene	1	2	3	4	5	6	7	8
Bayer reconstruction	R	30.71	30.71	28.85	28.81	27.12	30.03	27.48	23.48
	G	30.37	30.13	29.48	29.82	29.13	31.21	29.35	25.92
	B	30.79	29.97	28.87	28.89	27.65	30.01	28.57	23.38
	RGB	30.62	29.79	29.06	29.15	27.89	30.08	28.40	24.11
Proposed method	R	32.42	30.56	29.96	31.27	30.00	31.72	28.20	25.38
	G	33.14	31.38	30.83	31.95	31.39	32.81	29.90	26.11
	B	32.79	31.45	30.19	31.67	30.80	32.01	29.95	25.27
	RGB	32.78	31.11	30.31	31.62	30.70	32.16	29.27	25.57

	Scene	9	10	11	12	13	14	15	16
Bayer reconstruction	R	29.80	28.80	30.42	30.49	29.73	29.90	29.46	30.80
	G	29.97	29.39	30.19	30.27	30.04	30.17	30.14	30.33
	B	29.61	28.72	30.06	30.61	30.27	29.87	30.34	30.79
	RGB	29.79	28.96	30.22	30.45	30.01	29.98	29.96	30.64
Proposed method	R	31.93	30.18	32.44	32.57	30.90	31.71	30.45	32.30
	G	33.13	31.00	33.33	33.90	32.71	32.26	32.74	33.21
	B	31.77	30.28	31.99	32.64	32.03	31.72	31.95	32.38
	RGB	32.24	30.47	32.55	33.00	31.81	31.89	31.60	32.61

	Scene	17	18	19	20	Average
Bayer reconstruction	R	31.28	29.51	30.97	27.42	29.22
	G	30.57	30.00	30.47	29.21	29.76
	B	31.33	30.14	31.15	27.31	29.42
	RGB	31.04	29.87	30.85	27.90	29.44
Proposed method	R	33.36	30.77	32.09	31.07	30.96
	G	34.62	31.25	30.82	32.16	31.93
	B	33.38	31.87	31.69	30.95	31.34
	RGB	33.75	31.28	31.50	31.36	31.38

In Equation (10), S is the maximum pixel value, e.g., 255 for 8-bit images. Here, N is the total number of pixels in all frames. The value $\mathbf{f}_{true}(x,y,t)$ is the pixel value at position (x,y) of the time t reference image, and value $\mathbf{f}_{calc}(x,y,t)$ is the result of reconstruction.

We compared our method with the Bayer reconstruction [1,2,7-10,13,20] because they both use the same color filter array. Sensing images $\mathbf{g}$ are simulated by sampling the $\mathbf{f}_{true}$. We add the noise to $\mathbf{g}$ to be reflected during low

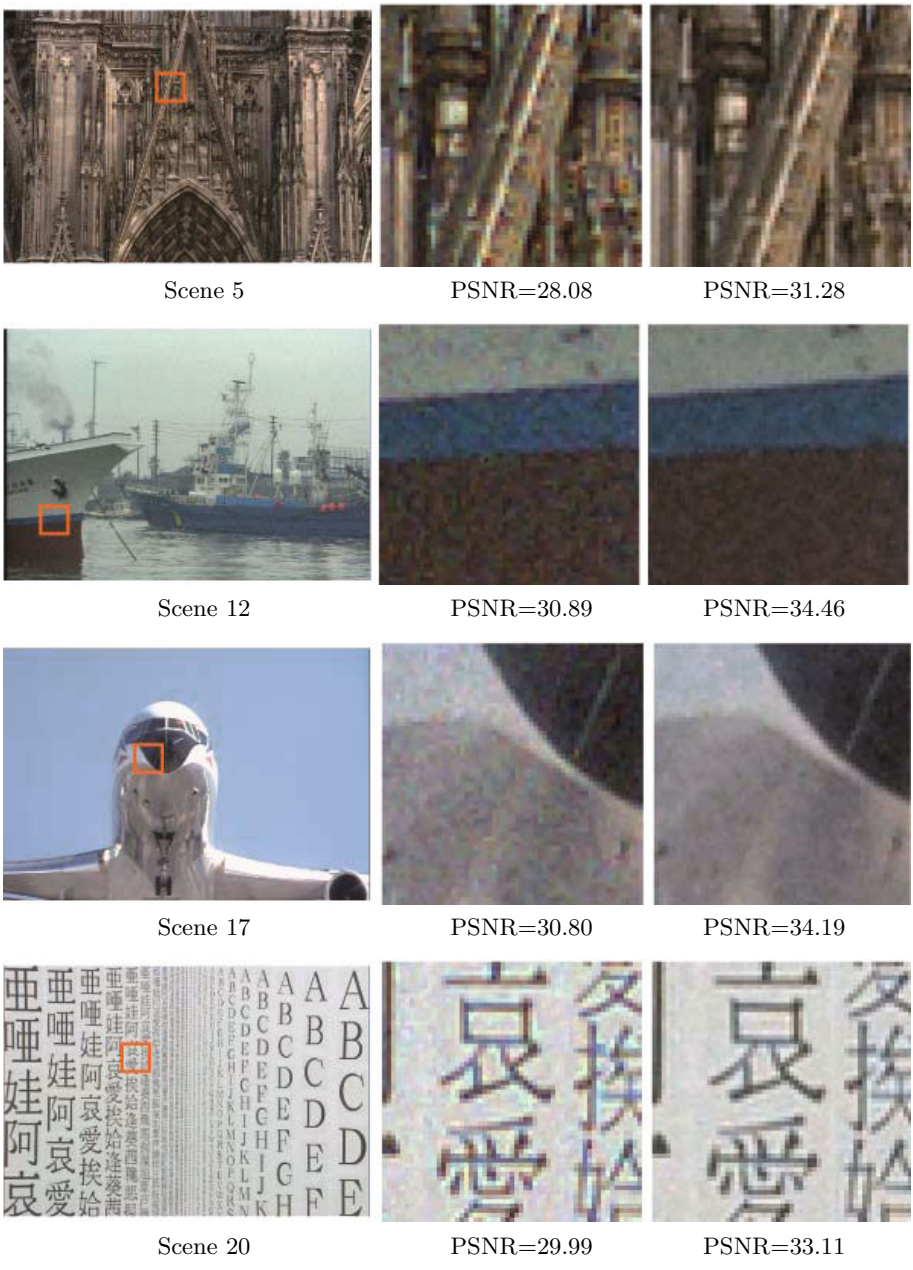


Fig. 4. (left) Test image sequences. (middle) Results of Bayer reconstruction and PSNR of picture area. (right) Results of proposed method and PSNR of picture area.

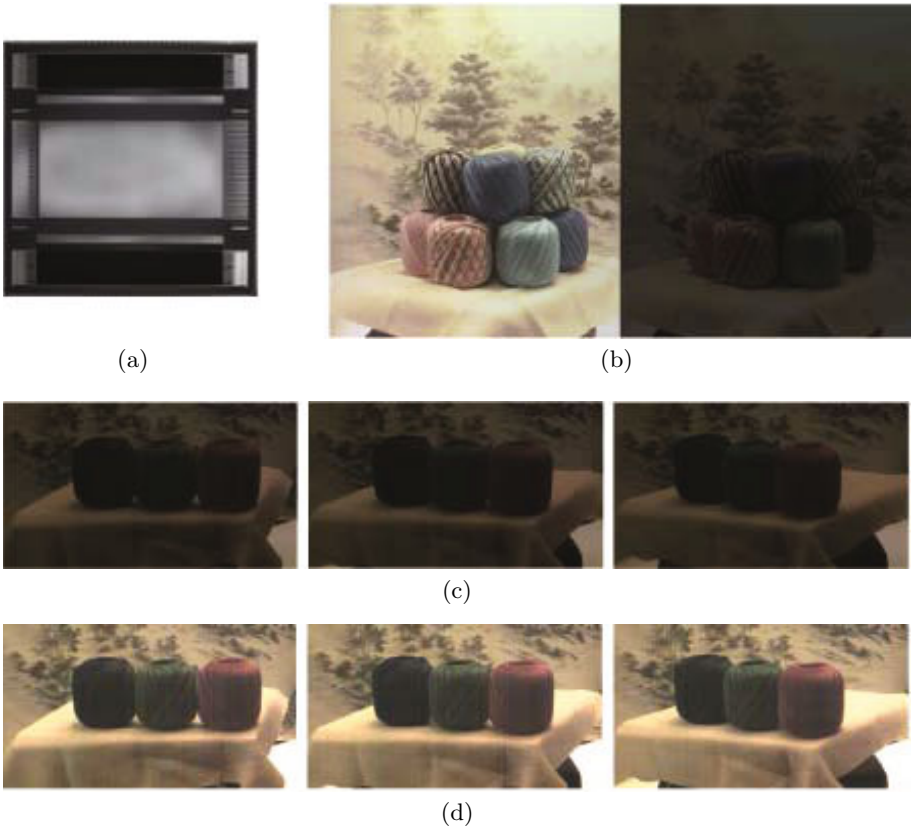


Fig. 5. (a) Image sensor used in proposed method. (b) Proposed method imaging and normal imaging. Proposed method uses long-term exposure for four frames. (c)Normal imaging sequence. (d) Proposed method imaging sequence.

illumination sensing. A long-term exposed image is defined as the sum of multi-frame images in **g**. The experimental conditions are listed in Table 1, and Table 2 and Figure 4 show the results of the experiment.

4.2 Development of Dedicated Image Sensor

We developed a dedicated image sensor to provide experimental proof. The single-chip image sensor is shown in Figure 5(a) and the results of using the sensor are shown in Figure 5(b)-(d).

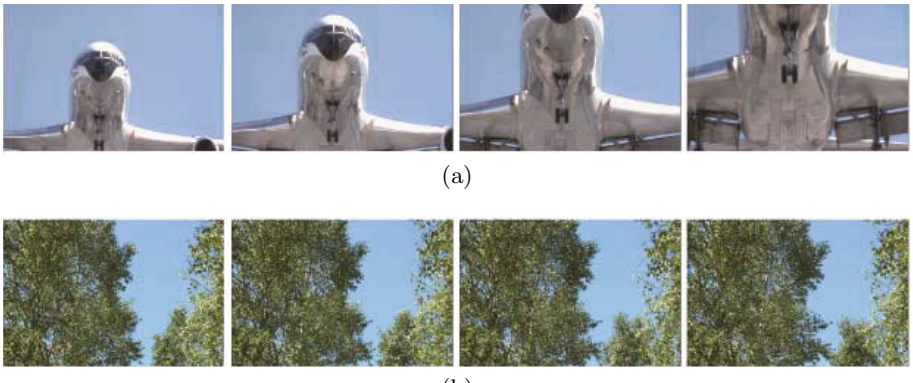
The image sensor is 4K2K CMOS image sensor and was fabricated using a 0.25-μm process. The 4K2K pixels composed of three transistors and a photo-diode are arranged in a Bayer pattern. The transfer gates in the R/B and G pixels are connected with the different row lines, respectively. To reduce the operation frequency, the vertical shift register drives two rows of pixels through the

multiplexer, which applies readout pulses, respectively. This circuit configuration enables the R/B and G pixels to the addressed independently.

5 Discussion

The simulation results showed that the PSNR of our method was about 2 dB higher than that gained with Bayer reconstruction. Furthermore, our proposed method worked effectively in low illumination conditions.

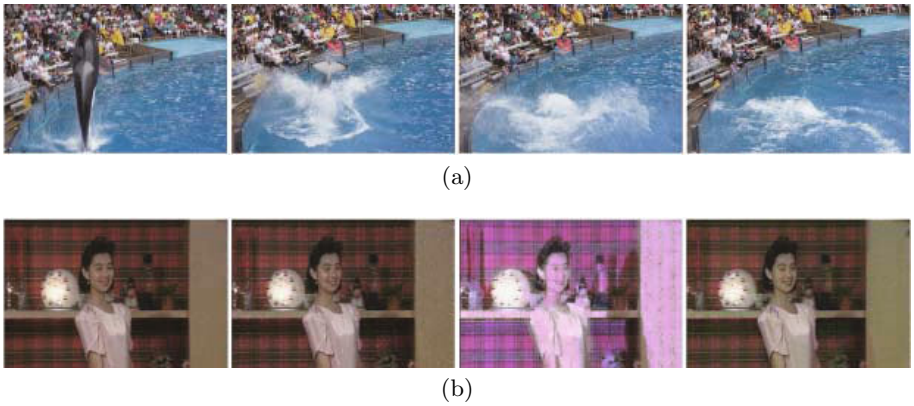
The PSNR changes for each scene because the motion detection accuracy differs. If the motion of the pixels are spatially smooth, the detection accuracy will be high because the proposed sensing method interpolates motion information of long-term exposed pixels. The simulation results of scene 17, which is depicted in Figure 6(a) show that the PSNR is the highest of the 20 scenes. Because the



(a)

(b)

Fig. 6. (a) Images of 30 frame intervals in scene 17. (b) Images of 30 frame intervals in scene 8.



(a)

(b)

Fig. 7. (a) Image sequence of scene 7. (b) Images sequence of scene 19.

scene 17 dominatly contains spatially smooth motions. On the other hand, as shown in Figure 6(b), for scene 8, the PSNR is the lowest of the 20 scenes. Because the scene 8 contains large of variety motions. The differences in the PSNR of our method and that of the Bayer reconstruction are small in scene 7 and scene 19. We think that one reason for this is that the accuracy of the motion detection of these scenes is low. As shown in Figure 7(a), for scene 7, the motion of both the dolphin and the spray of water is large. As shown in Figure 7(b), for scene 19, the value changes greatly between consecutive frames because the light sometimes flashes.

The super-resolution processing adapts from a low-resolution image sequence to a high-resolution image sequence. This method requires either motion of the object or motion of the camera. Because our method does not need either of these terms, it is more effective than the super-resolution processing.

As shown in Figure 5(b), because our method uses long-term exposure for four frames, sensitivity is increased by four times.

6 Concluding Remarks

We proposed the image reconstruction method and its dedicated sensor for high-sensitivity imaging. Experimental results showed that our proposed method is effective in conditions of low illumination. We developed a dedicated single-chip image sensor and increased the imaging sensitivity fourfold.

Acknowledgement. This work was supported in part by National Institute of Information and Communications Technology(NICT).

References

1. Adams Jr., J.E.: Design of practical color filter array interpolation algorithms for digital cameras. In: Proc. SPIE, vol. 3028, pp. 117–125 (1997)
2. Adams Jr., J.E.: Interactions between color plane interpolation and other image processing functions in electronic photography. In: Proc. SPIE, vol. 2416, pp. 144–151 (1995)
3. Agrawal, A., Gupta, M., Veeraraghavan, A., Narasimhan, S.G.: Optimal coded sampling for temporal super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition (2010)
4. Bayer, B.E.: Color imaging array. US. Patent 3, 971, 065 (1976)
5. Nayar, S.K., Ben-Ezra, M.: Motion-based motion deblurring. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(6), 689–698 (2004)
6. Bascle, B., Blake, A., Zisserman, A.: Motion Deblurring and Super-Resolution from an Image Sequence. In: Buxton, B.F., Cipolla, R. (eds.) ECCV 1996. LNCS, vol. 1065, pp. 573–581. Springer, Heidelberg (1996)
7. Cok, D.R.: Signal processing method and apparatus for producing interpolated chrominance values in a sampled color image signal. US Patent 4, 642, 678 (1987)
8. Hamilton Jr., J.F., Adams Jr., J.E.: Adaptive Color Plan Interpolation in Single Sensor Color Electronic Camera. US Patent 5, 506, 619 (1996)

9. Hamilton Jr., J.F., Adams Jr., J.E.: Adaptive color plane interpolation in single sensor color electronic camera. US Patent 5, 629, 734 (1997)
10. Hibbard, R.H.: Apparatus and method for adaptively interpolating a full color image utilizing luminance gradients. US Patent 5, 382, 976 (1995)
11. Honda, H., Iida, Y., Egawa, Y., Seki, H., Tanaka, N.: High Sensitivity Color CMOS Image Sensor with White-RGB Color Filter Array and Color Separation Process Using Edge Detection. In: International Image Sensor Workshop, pp. 263–266 (2007)
12. Iwabuchi, S., Maruyama, Y., Ohgishi, Y., Muramatsu, M., et al: A Back-Illuminated High-Sensitivity Small-Pixel Color CMOS Image Sensor with Flexible Layout of Metal Wiring. In: International Solid-State Circuits Conference, pp. 302–303 (2006)
13. Laroche, C.A., Prescott, M.A.: Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients. U.S.Patent 5, 373, 322 (1994)
14. Luo, G.: Color filter array with sparse color sampling crosses for mobile phone image sensors. In: International Image Sensor Workshop. pp. 162–165 (2007)
15. Popovic, Z.D., Sprague, R.A., Neville Connell, G.A.: Technique for monolithic fabrication of microlens arrays. Applied Optics 27(7), 1281–1284 (1988)
16. Rav-Acha, A., Peleg, S.: Two motion-blurred images are better than one. Pattern Recognition Letters 26(3), 311–317 (2005)
17. Rhodes, H., Tai, D., Qian, Y., Mao, D., et al.: The Mass Production of BSI CMOS Image Sensors. In: International Image Sensor Workshop, pp. 27–32 (2009)
18. Tai, Y., Du, H., Brown, M., Lin, S.: Image/Video Deblurring Using a Hybrid Camera. In: IEEE Conference on Computer Vision and Pattern Recognition (2008)
19. Wakabayashi, H., Yamaguchi, K., Okano, M., Kuramochi, S., et al.: A 1/2.3-inch 10.3Mpixel 50 frame/s Back-Illuminated CMOS Image Sensor. In: International Solid-State Circuits Conference (2010)
20. Weldy, J.A.: Optimized design for a single-sensor color electronic camera system. In: Proc. SPIE, vol. 1071, pp. 300–307 (1988)
21. Wuu, S.G.: BSI Technology with Bulk Si Wafer. In: International Image Sensor Workshop Symposium on Backside Illumination of Solid-State Image Sensors, pp. 124–153 (2009)
22. Yuan, L., Sun, J., Quan, L., Shum, H.Y.: Image deblurring with blurred/noisy image pairs. In: International Conference on Computer Graphics and Interactive Techniques, vol. 26(3) (2007)

On the Use of Implicit Shape Models for Recognition of Object Categories in 3D Data

Samuele Salti, Federico Tombari, and Luigi Di Stefano

Computer Vision Lab, DEIS, University of Bologna, Bologna, Italy
{samuele.salti,federico.tombari,luigi.distefano}@unibo.it
www.vision.deis.unibo.it

Abstract. The ability of recognizing object categories in 3D data is still an underdeveloped topic. This paper investigates on adopting Implicit Shape Models (ISMs) for 3D categorization, that, differently from current approaches, include also information on the geometrical structure of each object category. ISMs have been originally proposed for recognition and localization of categories in cluttered images. Modifications to allow for a correct deployment for 3D data are discussed. Moreover, we propose modifications to three design points within the structure of a standard ISM to enhance its effectiveness for the categorization of databases entries, either 3D or 2D: namely, codebook size and composition, codeword activation strategy and vote weight strategy. Experimental results on two standard 3D datasets allow us to discuss the positive impact of the proposed modifications as well as to show the performance in recognition accuracy yielded by our approach compared to the state of the art.

1 Introduction

Object categorization is among the most stimulating, yet challenging, computer vision tasks. It consists of automatically assigning a category to a particular object given its representation (an image, a point cloud, ..) and a predefined taxonomy. In the last decade the main effort has been devoted to categorizing classes of objects from images [1], one of the most prominent approaches being the application to image features of the Bag-of-Words paradigm, previously used for text categorization and document analysis. In particular, this approach, typically referred to as *Bag-of-Features* (BoF) or *Bag-of-Visual-Words* (BoVW), represents image categories as histograms ("bags") of feature descriptors [2,3,4]. To account for efficiency, histograms are not built on descriptors themselves but on an alphabet of descriptors, typically termed "codebook", obtained via clustering or vector quantization [1].

BoF methods turned out to be particularly effective even though, unlike some more recent proposals, they completely discard geometrical relationships between object parts. Among those leveraging geometric structure, one of the most successful proposals is Implicit Shape Models (ISM) [5], that encodes spatial relationships by means of a probabilistic Hough voting in a 3-dimensional

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 653–666, 2011.

space representing scale and translation. Moreover, the use of geometrically well-localized information allows these methods to be deployed also as detectors of specific object categories in presence of clutter, occlusion and multiple object instances. Typical object categories of interest have been pedestrians, faces, humans, cars [5].

The increasing availability of large databases of 3D models has pushed forward a growing interest towards computer vision and machine learning techniques capable of processing 3D point clouds and meshes. One of the most investigated tasks so far has been 3D object retrieval (see [6,7] for surveys) which aims at finding the most similar 3D models in the database to a given query model inputted by the user. Another well investigated topic concerns 3D object recognition [8,9]. Only very recently the first methods aimed at 3D object categorization have been proposed in literature. They mainly extend the BoF paradigm to the 3D scenario by representing categories as histograms of codewords obtained from local shape descriptions of 3D features [10,11,12].

In this paper we investigate on how to deploy Implicit Shape Modeling for the categorization of 3D data. Although in the reminder of this paper we will focus only on categorization, it is worth noting that this approach holds the potential to solve within the same framework the problem of simultaneous localization and classification of objects in cluttered scenes, even in presence of multiple instances.

2 3D Implicit Shape Model

The basic idea idea underlying Implicit Shape Models is to perform object category recognition and instances localization based on a non-parametric probability mass function of the position of the object center. These probability functions come from a probabilistic interpretation of the voting space of a Generalized Hough Transform algorithm. Votes are casted by local features that are matched against a codebook learned, together with votes, from a set of training examples. When applied to 3D data, we identify the general form of an algorithm training a 3D ISM as follows (Fig. 1):

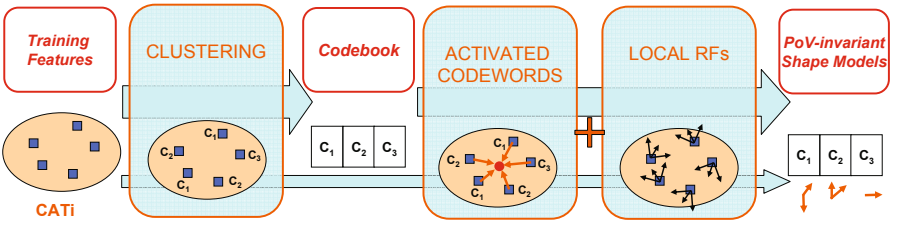


Fig. 1. Overview of the training stage of 3D ISM

- local features are detected and described from the 3D training data.
- for each category C_i
  - all features belonging to C_i are clustered to create the codebook of C_i
  - for each training feature $f_j^{C_i}$ of category C_i
    - $f_j^{C_i}$ is matched against the codebook of C_i according to a *codeword activation strategy*.
    - each activated codeword adds to the ISM of C_i the position of $f_j^{C_i}$ with respect to the object center. Each feature $f_j^{C_i}$ needs to incorporate a repeatable local Reference Frame (RF), and votes are expressed with respect to such local RF of $f_j^{C_i}$.

Then, a generical 3D ISM recognition procedure may be decomposed in the following steps (Fig. 2):

- local features are extracted and described from the 3D input data.
- for each feature f_j and each category C_i
  - f_j is matched against the codebook of C_i according to a *codeword activation strategy*.
  - each activated codeword casts its set of votes for the Hough Space of C_i in its ISM.
  - votes are rotated and translated so as to be expressed in the local RF of the input features before voting, thus obtaining *Point-of-View (PoV) independent votes*. The magnitude of the vote is set according to a *vote weighting strategy*.
- in case of categorization of 3D database entries, the category yielding the global maximum among all the Hough spaces is selected as output; in case of detection in a cluttered scene, local maxima of each category above a threshold are selected as category instance hypotheses for a further verification stage and/or pose estimation.

This scheme exhibits two main differences with respect to the use of ISM for detection of object categories in 2D images. First of all, since the sensor produces metric data, there is no need for scale invariance: in the 2D case, when casting votes for the object center, the object scale is treated as a third dimension in the voting space. With 3D data we can cast votes for object hypotheses directly in the coordinates space, which is again a 3D dimensional space. The second difference regards the use of PoV-independent votes, that leads to a PoV-independent detector. In the original ISM proposal, objects of the same category being seen under different point of views are regarded as instances of different, unrelated categories. It is worth pointing out that the use of PoV-independent votes is not just a nice extension that allows for more flexibility of the final method, it is indeed mandatory when using 3D ISM to categorizes 3D database entries, for these cannot be assumed to be expressed within the same global RF. Unfortunately, most of the proposals in the field of 3D local features do not include a fully defined local RF, *e.g.* Spin Image [8] uses just one repeatable axis, the normal, and 3D Shape Context [9] uses a random, not

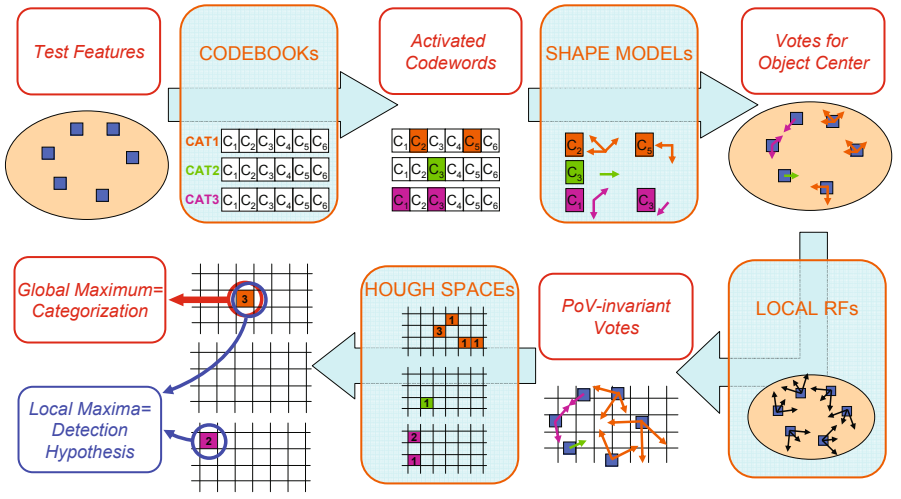


Fig. 2. Overview of 3D ISM for Categorization and Detection

repeatable direction on the tangent plane to define a full 3D local RF. However, SHOT [13] is a recent 3D descriptor proposal that includes a repeatable local RF and yields state-of-the-art performance. We thus use these features throughout this work. In turn, one of the contribution of this paper is to show that such recently proposed features demonstrate good performance even in 3D object categorization, an experiment that was not proposed in [13].

In the previous overview of the method we have highlighted the main design decisions that need to be taken to define a 3D ISM, i.e. the *codeword activation strategy* and the *vote weighting strategy*. In the following we address, by discussion and experiments, the possible alternatives for these design choices together with other major issues related to *codebook size and composition*. It is worth noting that, although we have conducted experiments using 3D data only, all our reasoning is independent from data dimensionality. Therefore, we expect the observations drawn from our analysis to be beneficial also for the case of standard 2D ISMs.

3 Codebook

3.1 Codebook Size

Codebooks are widely used for 2D and 3D object categorization (e.g. [14] [10] [11]). The reason behind their use is efficiency, both in terms of memory occupancy of the codebook and computational time for codeword activation. They are not expected to have any positive impact on the generalization abilities of the algorithms. They are usually built by applying some standard clustering algorithms, like K-Means, on the features extracted from the training data. Little attention, however, has been paid to the loss in discriminative power of the

Fig. 3. Impact of codebook size on mean recognition rate and mean recognition time

codebook after size reduction. Furthermore, research in the field of Approximate Nearest Neighbor provides us efficient methods to solve the codeword activation problem even in high dimensional spaces and with large databases [15]. Finally, the cost of storing a set of descriptors for each training model of the currently publicly available 3D datasets is nowadays definitely affordable by off-the-shelf machines. Based on the above considerations, we investigated on the actual importance of building a codebook to successfully perform object category recognition in 3D data.

The chart in Fig. 3 shows the outcome of an experiment carried out on the Aim@Shape Watertight dataset (see Sec. 6 for more details about the dataset and the experimental methodology). We used half dataset for training and half for testing, i.e. ten models for training and ten for testing for each category. 200 mesh vertexes were randomly selected on each training model obtaining 2000 features as training set for each category. We then performed K-Means on this set, varying K logarithmically from 10 to 2000. We used such codebooks to categorize the test set. The best mean recognition rate is obtained with 2000 codewords, i.e. using the plain training data without any clustering. Loss in efficiency is minimal, for instance using 100 codewords the mean time to categorize one test model is about 42 ms, whereas using the plain training set as codebook it slightly increases to about 52 ms. Memory occupancy, of course, scales linearly with codebook size and, for the considered dataset, when using no clustering is less than 57MB. Therefore, based on the indication of this and other similar experiments, in the following we use as "codebook" the whole training data, without carrying out any clustering on them.

3.2 Sharing Codewords among Categories

In the original ISM proposal, the case of simultaneous recognition of multiple categories is solved by running a detector for each category, endowed with its own codebook built from training data belonging to its category. We refer to this configuration as ISM with *separated codebooks*: codebooks of different

categories are independently built and used. In the context of categorization of DB entries, we have investigated on another possible configuration, that we refer to here as ISM with *global codebook*: a codebook is created from the training data belonging to all categories and then used by all ISMs. The Shape Model of each category is still built during the training stage by considering only the training data belonging to that category. However, denoting with SM_i the Shape Model of category C_i, not only those originated by the training data of C_i, but all the codewords in the codebook, regardless of the categories of the features that generated them, can participate to SM_i, provided that they are similar - according to the codeword activation strategy - to any of the training features of C_i. Therefore, this scheme endows the ISM paradigm with a broader capability of generalization: whilst the separated codebooks configuration is able to generalize at an intra-class level, by letting features observed in different training instances of the same class collaborate to the detection of an instance during testing, the global codebook configuration lets ISM generalize also at an *inter-class* level. It allows features observed in training examples of different categories to reinforce the hypothesis that an instance of category C_i is present. In other words, it builds a "universal" codebook of all the likely features given the training data, and then associates a spatial location for a specific category to all those that are "similar" to the training features of such category, regardless of the labels of the training data that originated that codeword.

It is worth highlighting that memory requirements of both configurations are equal: although a global codebook requires C times more space than a separated codebook, with C the number of categories, only one instance of it has to be stored in memory since it can be shared among all the C 3D ISM required by our proposal. Query time scales logarithmically with the size of the codebook: since codewords in the global codebook are C times those of the separated codebooks, query time is increased by $\log C$, a limited amount for typical number of categories in publicly available 3D databases (i.e. less than 30).

4 Codeword Activation Strategy

The codeword activation strategy proposed for the deployment of ISM in the case of 2D data [5] is the *cutoff threshold*: codewords are activated, and, thus, cast their votes, if their distance from the test feature is below a threshold. An alternative approach is represented by the k-NN activation strategy: the closest k codewords to the test feature are activated, regardless of their distance. We consider the latter strategy more suitable to the task of categorization, the reason being twofold. First of all, in those parts of the feature space characterised by a high codeword density, k-NN activates generally less features than the cutoff strategy, only the k most similar ones. By increasing the number of votes casted by each test feature in the Hough Space we may expect to sharpen the peak corresponding to a true instance of the class, but also to generate spurious peaks in the voting space, by randomly accumulating wrong votes in the same bin. In such parts of the feature space, the k-NN strategy acts as a filter that aims at

reducing the probability of adding noise into the Hough Space, while it hopefully retains the ability to let the correct hypothesis emerge, by selecting only the most similar codewords. Secondly, in those parts of the feature space with a low density or even absence of codewords, k-NN activates anyhow k codewords, whereas the cutoff strategy cast very few votes, if any. Indeed, being the threshold generally chosen as small as to prevent generation of false peaks, the cutoff strategy generally tends not to activate any codeword in low density regions of the feature space. Obviously, the codewords activated by the k-NN strategy can be really different from the test data. Still, given the training set, they are the most similar available: if we have to generalize from the training examples to attempt to classify the current input, they appear a reasonable choice. The same reasoning does not hold when using 3D ISM to detect instances in cluttered scenes: in such a case, a high distance from any codeword is likely to indicate that the test feature comes from clutter and hence should not cast votes, such behavior being correctly modeled by the cutoff strategy. Yet, when reasoning in absence of clutter, as it is the case of categorization of entries of a 3D database, the k-NN strategy offers an adaptive behavior with respect to the training data that seems more suitable to the task.

5 Votes Weighting Strategy

In [5], the vote weight for each pair (test feature, vector in the shape model) is given by the product of a match weight and an occurrence weight

$$w = p\left(o_n, x|C_i, l\right) * p\left(C_i^*|f_k\right) = \frac{1}{|M|} * \frac{1}{|Occ\left[i\right]|} \tag{1}$$

with M being the set of codewords activated by the test feature f_k and $Occ[i]$ being the set of vectors in the Shape Model associated with codeword i.

The rationale behind this choice is tightly coupled with the use of the original ISM for detection in cluttered scenes. In presence of clutter, there is an obvious trade off between increasing the number of true detections and limiting the number of false detections. The choice of the vote weighting strategy operated in [5] goes in this direction. If a feature activates more codewords than another feature and/or if such codewords can be observed in more feasible positions with respect to the object center than other codewords, then this feature will be regarded as less distinctive since it likely generates more spurious votes in the Hough Space. By keeping low the weight, i.e. the confidence, on the position of the object center for the votes of such features, the original ISM tries to choose a good working point to optimize the above mentioned trade-off, by keeping below the detection threshold such spurious local maxima of the voting space. We refer to this vote weighting strategy as *Localization Weights (LW)*.

Again, in absence of clutter the scenario is different. Recall from Sec. 2 that we propose to select as output the category yielding the global maximum among all the Hough spaces. Therefore, in this case the emphasis for each 3D ISM should be on supporting as much as possible its best hypothesis. This means that spurious

local maxima are not relevant for categorization, as long as they do not hide the true global maximum. Since we can reasonably expect that the geometrical consistent bin will likely be the strongest peak in the voting space, there is no reason to try to weaken local maxima by acting on the vote weight. On the other hand, using the original ISM vote weighting strategy may uselessly reduce the strength of the global maximum only because features that casted vote for it have also casted votes for wrong locations, and this can lead to a wrong selection of the correct category in the final competition among each global maximum of all categories. Hence, in the case of categorization, we have investigated on the use of the same constant weight for all features and codewords. Hereinafter, we will denote this vote weighting strategy as *Categorization Weights (CW)*.

6 Experimental Results

We have tested our proposals on the Aim@Shape Watertight (ASW) dataset, previously used for the evaluation of 3D object categorization algorithms such as [10], and on the Princeton Shape Benchmark (PSB) [16], already used for 3D categorization in [11]. Since meshes in the PSB dataset exhibit a high variance in metric dimensions, even within the same class, to define a Hough Space suitable for all meshes, we normalize models before using them for testing or training. Specifically, we translate the model barycenter into the origin, compute the Eigenvalue Decomposition (EVD) of the scatter matrix of each model to find its principal axes, we scale the model down or up by a scale factor given by $1/X_{max} - X_{min}$, with X_{max},X_{min} the maximum and minimum coordinates of the mesh along the first principal axis, and finally rotate the model to align it with its principal axes. It is important to note that, due to the sign ambiguity inherent to the EVD (see e.g. [13]), we still need PoV-independent votes to achieve correct categorization. This normalization allows also for an important simplification: we can define the Hough Space just around the barycenter, i.e. the origin: any hypothesis for the object center laying far away from the barycenter will clearly be a spurious peak in the voting space. This improves both the effectiveness and the efficiency of our method, since it reduces the memory footprint needed to store the Hough Space. In particular, we used a Hough Space consisting of one squared bin, centered in the origin and with a side of 0.2. In all the experiments with both datasets we randomly extract 200 feature points from each training model and 1000 feature points from each testing model, and we describe them using SHOT with 16 spatial sectors (8 on the tangent plane and 2 concentric spheres) and 10 bins for the normal histograms. We do not perform any multi scale description, we use just a single support radius, equal to 0.25 and 0.45 for the AWS and the PSB dataset, respectively. As discussed in section 3.1, we use a plain codebook composed by all training descriptors.

The Aim@Shape Watertight dataset contains 20 categories, each composed of 20 models. We tested our performance on this dataset according to two methodologies. First, we divided the dataset in a training and a testing set by taking the first 10 models of each category as training set and the rest as testing set. With

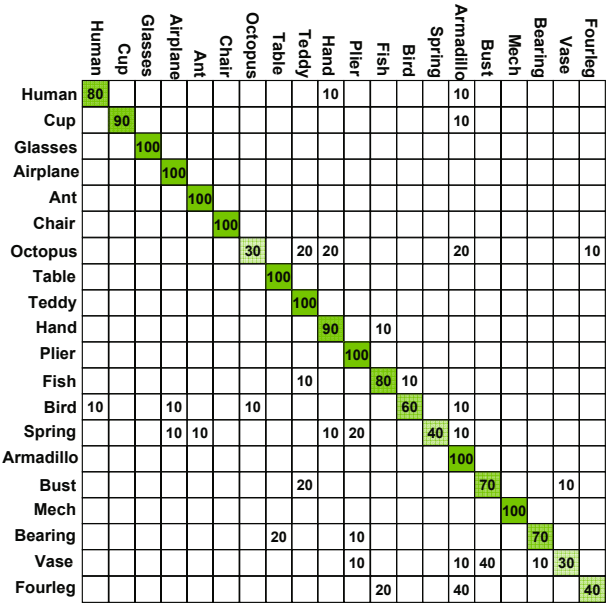


Fig. 4. Confusion Matrix for Aim@Shape Watertight, 1-NN Codeword Activation Strategy and CW Votes Weighting Strategy. The rows represent the test categories of the input model, the columns the output of the 3D ISM.

this configuration we studied the influence of the previously discussed design issues. Then, we also performed Leave-One-Out cross validation as done in [10], to be able to compare our results with such related work. Of course, the first test is more challenging, since significantly less training data is available to learn category shapes.

Results for the first series of experiments are reported in Fig. 5. We compared the performance of all the combinations of the proposed design decisions, i.e. global codebook (GC) vs. separated codebooks (SC), LW vs. CW and k-NN vs. cutoff with different values. The best recognition rate for this dataset is 79% and is obtained using 1-NN as Codeword Activation Strategy and a global codebook. In such configuration LW is the same as CW, since each codeword has zero or one vote. Fig. 4 reports the confusion matrix for such case.

In the case of the Leave-One-Out cross validation, [10] reports a mean recognition rate of 87.25%. Using 2-NN as Codeword Activation Strategy, a global codebook and CW as Votes Weighting Strategy, we have obtained **100**%.

The PSB dataset comes with a hierarchical categorization and a predefined division in training and testing sets. We use such categorization and such division. To compare our results against those in [11] we use the categorization level named Coarse 2, although it defines quite abstract meta-categories, such as "Household", which includes electric guitars, guns as well as stairs, or "-1", that stands for "all other models in the dataset". Clearly this dataset is more challenging than ASW, the intra-class and the inter-class variability being definitely higher.

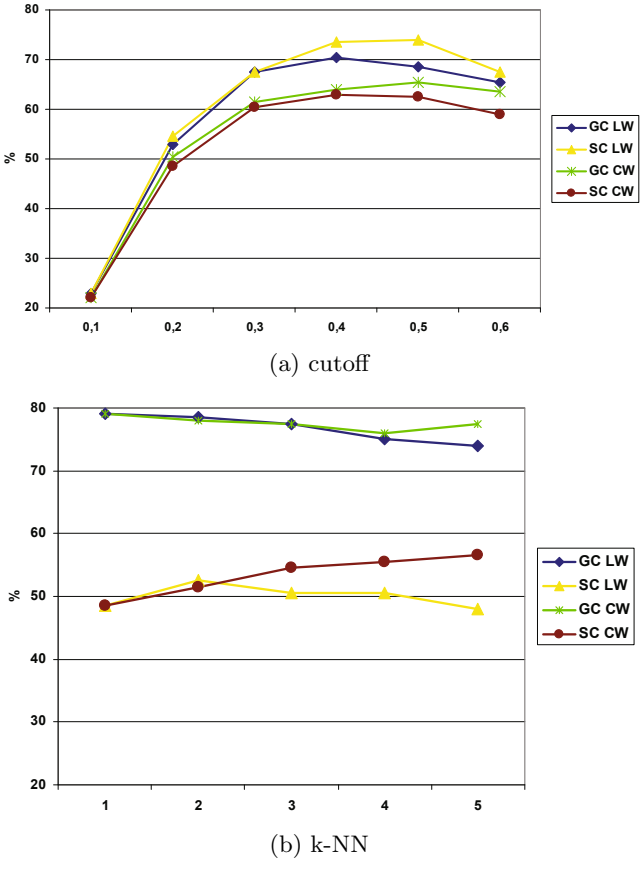


Fig. 5. Mean recognition rate as a function of varying cutoff and k-NN values on Aim@Shape Watertight

Results are reported in Fig. 6. We compared the same combinations as in the previous experiment. The best recognition rate for this dataset is 50.2% and is obtained using 2-NN as Codeword Activation Strategy, a global codebook and the CW Votes Weighting Strategy. [11] reports a mean recognition rate of 55%. It is worth noting that, in addition to the previously mentioned difficulties, the PSB dataset presents also a highly variable point density among the models. As it has been noted in [13], point density variation is not well tolerated by current 3D descriptors. This was explicitly accounted for in [11], where all PSB meshes were resampled to a constant number of vertexes, uniformly distributed in the meshes. We have not implemented such resampling yet, that could likely improve our performance.

7 Discussion

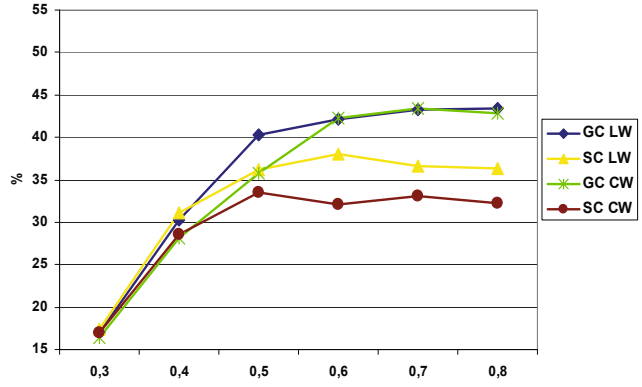
The most evident outcome of our investigation is definitely the fact that the Codeword Activation Strategy and codebook composition play a significant role on

the performance of 3D ISM for categorization. In both datasets k-NN with global codebook consistently outperforms the cutoff threshold with both kinds of codebook composition, regardless of the choice of k. This confirm two intuitions: a) that the intrinsic adaptation to codewords density in the feature space provided by k-NN is more suitable for database entries categorization, i.e. in absence of clutter, since it enhances ISM generalization ability; b) that the global codebook, when compatible with the application constraints on memory occupancy and computation time, endows ISM with higher, inter-class generalization power.

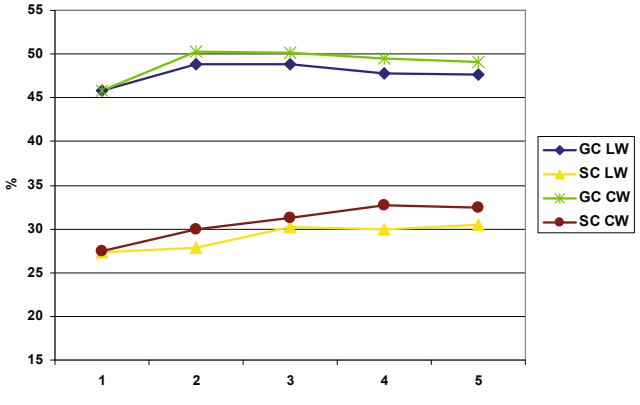
Experiments also reveal a tight coupling between the use of k-NN and the global codebook: k-NN with separated codebooks exhibits unsatisfactory performance, even with respect to the cutoff strategy. With the global codebook the k nearest neighbor codewords for a test feature are the same for each tested category, i.e. they represent the overall k most similar features throughout those belonging to all categories seen in the training stage, what then differs for the different categories is how these codewords vote in the different ISMs. In particular, it is worth pointing out that, differently from the case of separated codebooks, it happens that some of the codewords have no associated votes in the ISM of a specific category. This happens when a codeword is not similar to any training data of that category. Therefore, many of the k activated codewords will likely vote only for a subset of the categories, so that votes accumulation in the Hough Space has more chances to let the true category emerge, being required to filter out a limited amount of wrong votes. In other words, this configuration balances the impact of codebook (i.e. of features similarity) and shape model (i.e. of geometrical structure) and results in good recognition rates. With separated codebooks, instead, the k nearest neighbors are different in different codebooks, so that in several of them the activated codewords may be very dissimilar to the test feature. Moreover, since there are no codewords without votes in this configuration, all the activated codewords will cast votes in their shape models. This configuration, therefore, tends to diminish the importance of feature similarity and relies almost completely on shape models being able to select the correct category. This increases the probability of generating wrong, spurious peaks in the voting space.

The vote weighting strategy does not play a role as important as the other two discussed design decisions. Nevertheless, as far as the k-NN codeword activation strategy is concerned, the Categorization Voting obtains consistently slightly better performance in both datasets and with both kind of codebooks. This provides experimental evidence to the reasoning of Sec. 5.

As for the experiments on the cutoff threshold strategy, whilst on the PSB dataset the global codebook is still the favorable option, and there is little difference between the votes weighting strategies, in the case of the ASW dataset the decisive factor for obtaining higher performance seems to be the LW strategy whereas, unlike in the k-NN case, the codebook options seem to have quite a minor impact. We ascribe the latter to the cutoff strategy intrisecally balancing feature similarity and geometrical structure, for dissimilar codewords, given the cutoff threshold, cannot cast votes at recognition time also when the separated



(a) cutoff



(b) k-NN

Fig. 6. Mean recognition rate as a function of varying cutoff and k-NN values on the PSB coarse 2 dataset

codebook is used. On the other hand, it is quite more difficult to explain the higher performance of LW on this dataset. The higher performance of LW seems to suggest that in the ASW dataset wrong categories are supported in the voting space by less distinctive codewords, whose vote weights are indeed diminished by using LW.

The Confusion Matrix in Fig. 4 evidences how, beside gross errors that must be accounted to the difficulty of the task, several errors are somehow reasonable for an algorithm that tries to categorize objects based only on 3D shape. For instance, the category "Octopus", for which our proposal fails to recognize the majority of test models, is confused with "Hand", "Armadillo" and "Fourleg", i.e. with categories that present sort of "limbs" in configurations similar to those assumed by the models in the "Octopus" category. The 40% of "Fourleg" test models are wrongly categorized as "Armadillo", which, again, in some training models appears in a Fourleg-like pose. All the wrongly assigned test models of

"Bearing" are labeled as "Table" or "Plier", which have parts (the legs, the handles) that are shaped as bearings. Provided that this dataset can be successfully categorized by using only shape when enough training data can be deployed, as our 100% result in the Leave-One-Out test demonstrates, the mostly reasonable errors in the Confusion Matrix show that our proposal is able to learn a plausible, although less specific, model for the category shape in presence of less training data.

8 Conclusions

We have presented a new proposal for categorization of 3D data, which relies on the deployment of Implicit Shape Models in combination with a recently proposed 3D descriptor. We have devised the general structure of a 3D ISM and, based on its analysis, identified and discussed three design decisions that could improve the performance of the method when used for categorization. Experimental results on two well known and relative large datasets demonstrate that the combination of the k-NN codeword activation strategy and the use of a global codebook built from the whole training data of all categories is more suited to categorization than the standard ISM approach. Votes weighting strategy, on the other hand, does not seem to play such an important role for overall performance. The proposed optimal configuration compares favorably with the state of the art in 3D data categorization, obtaining similar results in one case and outperforming current proposals on the other considered dataset.

References

1. Pinz, A.: Object categorization. Foundation and Trends in Computer Graphics and Vision 1, 255–353 (2005)
2. Csurka, G., Bray, C., Dance, C., Fan, L.: Visual categorization with bags of keypoints. In: ECCV Workshop on Stat. Learning in Computer Vision (2004)
3. Sivic, J., Russell, B., Elfros, A., Zisserman, Z.: Discovering objects and their location in images. In: Proc. ICCV (2005)
4. Serre, T., Wolf, L., Poggio, T.: A new biologically motivated framework for robust object recognition. In: Proc. CVPR (2005)
5. Leibe, B., Leonardis, A., Schiele, B.: Combined object categorization and segmentation with an Implicit Shape Model. In: Proc. ECCV, pp. 17–32 (2004)
6. Tangelder, J.W.H., Veltkamp, R.C.: A survey of content based 3D shape retrieval methods. In: Proc. Shape Modeling International, pp. 145–156 (2004)
7. Iyer, M., Jayanti, S., Lou, K., Kalyanaraman, Y., Ramani, K.: Three dimensional shape searching: state-of-the-art review and future trends. Computer Aided Design 5, 509–530 (2005)
8. Johnson, A., Hebert, M.: Using spin images for efficient object recognition in cluttered 3D scenes. PAMI 21, 433–449 (1999)
9. Frome, A., Huber, D., Kolluri, R., Bülow, 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)

10. Toldo, R., Castellani, U., Fusiello, A.: A *bag of words* approach for 3D object categorization. In: Gagalowicz, A., Philips, W. (eds.) MIRAGE 2009. LNCS, vol. 5496, pp. 116–127. Springer, Heidelberg (2009)
11. Liu, Y., Zha, H., Qin, H.: Shape Topics: a compact representation and new algorithms for 3D partial shape retrieval. In: Proc. CVPR (2006)
12. Ohbuchi, R., Osada, K., Furuya, T., Banno, T.: Salient local visual features for shape-based 3D model retrieval. In: Proc. Int. Conf. on Shape Modeling and Applications, pp. 93–102 (2008)
13. Tombari, F., Salti, S., Di Stefano, L.: Unique signatures of histograms for local surface description. In: Proc. European Conference on Computer vision (ECCV 2010) (2010)
14. Sivic, J., Zisserman, A.: Video google: Efficient visual search of videos. In: Ponce, J., Hebert, M., Schmid, C., Zisserman, A. (eds.) Toward Category-Level Object Recognition. LNCS, vol. 4170, pp. 127–144. Springer, Heidelberg (2006)
15. Muja, M., Lowe, D.G.: Fast approximate nearest neighbors with automatic algorithm configuration. In: International Conference on Computer Vision Theory and Application VISSAPP 2009, pp. 331–340. INSTICC Press (2009)
16. Shilane, P., Min, P., Kazhdan, M., Funkhouser, T.: The Princeton Shape Benchmark. In: Shape Modeling International (2004)

Phase Registration of a Single Quasi-Periodic Signal Using Self Dynamic Time Warping

Yasushi Makihara[1], Ngo Thanh Trung[1], Hajime Nagahara[2], Ryusuke Sagawa[3], Yasuhiro Mukaigawa[1], and Yasushi Yagi[1]

[1] Osaka University, 8-1 Mihogaoka, Ibaraki, Osaka, 567-0047, Japan
[2] Kyushu University, 744, Motooka, Nishiku, Fukuoka, 819-0395, Japan
[3] AIST, 1-1-1 Umezono, Tsukuba, Ibaraki, 305-8568, Japan

Abstract. This paper proposes a method for phase registration of a single non-parametric quasi-periodic signal. After a short-term period has been detected for each sample by normalized autocorrelation, Self Dynamic Time Warping (Self DTW) between a quasi-periodic signal and that with multiple-period shifts is applied to obtain corresponding samples of the same phase. A phase sequence is finally estimated by the optimization framework including the data term derived from the correspondences, the regularization term derived from short-term periods, and a monotonic increasing constraint of the phase. Experiments on quasi-periodic signals from both simulated and real data show the effectiveness of the proposed method.

1 Introduction

Periodic signal analysis has been widely studied in the computer vision field as well as signal processing field, as the periodic signal plays quite an important role in many applications ranging from transmitting information via a radio carrier wave [1] [2] in the electronic communication field to periodic motion detection from video, periodic action recognition (e.g., walking and running), person identification from periodic motion (e.g., gait-based person identification [3]).

Such a periodic signal is often modulated in terms of amplitude, frequency, and phase by design or by chance, and is converted into a *quasi-periodic signal*. Typical examples of intentional modulation are Amplitude Modulation (AM) and Frequency Modulation (FM) [1] used in radio broadcasts, and Phase Modulation (PM) [2] used in radio control, where a carrier wave with known parameters is given as reference and the modulation is estimated based on the carrier wave.

On the other hand, accidental modulation is induced by a fluctuation in the sampling interval (network camera with limited communication band width) or that of the periodic signal source itself (e.g., fluctuations in human walking patterns). Estimating phases from such phase-modulated quasi-periodic signals is quite an important task in many applications. For example, temporal interpolation of a video with constant phase evolution needs the correct phase information for each key frame. Moreover, temporal super resolution of a periodic image sequence needs accurate phase registration data with sub-sampling order

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 667–678, 2011.

displacement of phase, in the same way that spatial super resolution needs image registration data with sub-pixel order displacement [4]. Phase registration data is also essential to reconstruct a manifold parameterized by phase in periodic action analysis and recognition and accurate period segmentation for periodic signal matching. In cases where a reference periodic signal is available, Dynamic Time Warping (DTW) [5] (more specifically, continuous DP [6] in the periodic signal case) is a powerful tool for matching two sequences with non-linear time warping, in the sense that matching results give phase registration data. The reference signal is, however, usually not available in the above applications.

This paper tackles the challenging problem of phase registration from a *single quasi-periodic* signal. After a short-term period has been detected for each sample, Self DTW between the quasi-periodic signal and that with multiple-period shifts is applied to obtain corresponding samples with the same phase. A phase sequence is finally estimated in a sub-sampling order by the optimization framework where an objective function is composed of the data term derived from the correspondences, the regularization term derived from short-term periods, and a monotonic increasing constraint of the phase.

2 Related Work

Parametric representation: A periodic signal is usually represented by a periodic function parameterized by amplitude, frequency, and phase, and it is often observed together with additive noise. Such parametric expression is widely used in the context of periodic signal reconstruction [7] and detection [8], enhancement of a specific frequency [9], estimation of amplitude [10], and decomposition of multiple periodic signals [11] [12] [13]. The common key technique in these approaches is parameter estimation and hence, non-parametric periodic signals are out of scope.

Linear time warping: Linear time warping is conventionally used in periodic action recognition such as gait recognition [14] [15] [16]. Periods are usually first detected as an interval of signal peaks [3] by maximum entropy spectrum estimation [17] or by maximum normalized autocorrelation [18]. The signals are then linearly stretched/shrunk so that the periods of two signals match. Naturally, these methods cannot deal with non-linear time warping within a period.

Non-linear time warping: Dynamic Time Warping (DTW) [5] has been widely used for elastic matching of two sequences in the field of action recognition [19] and gait recognition [20]. The Hidden Markov Model (HMM) is a probabilistic framework version of the DTW, which is also used in phase state estimation in walker motion extraction [21], gait silhouette refinement [22] [23], and gait recognition [24] [25]. An HMM needs sufficient training sequences and hence, cannot be applied directly to phase registration from a single sequence. Moreover, the number of states should be sufficiently large to realize a sub-sampling order phase estimation and this leads to an explosive increase in the number of training samples required.

3 Phase Registration

3.1 Problem Statement

Given a periodic function of the multi-dimensional signal $\boldsymbol{f}(t)$ with period P that satisfies $\boldsymbol{f}(t + jP) = \boldsymbol{f}(t)\ \forall j \in \mathbb{Z}$, a time normalized by period P, is introduced as an absolute phase s and a relative phase $\tilde{s}$ as

$$s = s_P(t) = \frac{1}{P}t \tag{1}$$

$$\tilde{s} = s - \lfloor s \rfloor, \tag{2}$$

where $s_P(t)$ is a phase function, and $\lfloor s \rfloor$ is a floor function. A normalized periodic function is subsequently introduced as

$$\boldsymbol{h}(s) = \boldsymbol{f}(s_P^{-1}(s)), \tag{3}$$

which satisfies $\boldsymbol{h}(s) = \boldsymbol{h}(\tilde{s})\ \forall s$.

Next, it is assumed that the phase function $s_P(t)$ is distorted by fluctuation into $s_Q(t)$ and that the periodic signal $\boldsymbol{f}(t)$ is converted to a quasi-periodic signal $\boldsymbol{g}(t)$, which is subject to

$$\boldsymbol{g}(t) = \boldsymbol{h}(s_Q(t)) = \boldsymbol{f}(s_Q(s_P^{-1}(s))). \tag{4}$$

Given the quasi-periodic signal $\boldsymbol{g}(t)$ and its phase function $s_Q(t)$, the periodic function is reconstructed as

$$\boldsymbol{h}(s) = \boldsymbol{g}(s_Q^{-1}(s)). \tag{5}$$

In addition, since the signal is usually quantized in observation, we redefine the above variables at quantized time t_i $(i = 0, \ldots, N)$ with subscription i (e.g., $\boldsymbol{g}_i = \boldsymbol{g}(t_i)$). Therefore, our objective is to estimate a phase sequence $\boldsymbol{S}_Q = \{s_{Q,i}\}$ from a given quasi periodic sequence $\boldsymbol{G} = \{\boldsymbol{g}_i\}$. This is referred to as the *phase registration* problem in this paper.

On the other hand, the following ambiguity of the phase function and normalized periodic function remains. Given another phase function $s'_Q(t)$ and another normalized periodic function $\boldsymbol{h}'(s) = \boldsymbol{h}(s'_Q(s_Q^{-1}(s)))$ that satisfies $\boldsymbol{h}'(s) = \boldsymbol{h}'(\tilde{s})\ \forall s$, another quasi-periodic function $\boldsymbol{g}'(t)$ is defined in two ways as $\boldsymbol{g}'(t) = \boldsymbol{h}(s'_Q(t)) = \boldsymbol{h}'(s_Q(t))$. Therefore, given the quasi-periodic function $\boldsymbol{g}(t)$, the ambiguity of combinations of the phase function $s_Q(t)$ and the normalized periodic function $\boldsymbol{h}(s)$ remains. In this paper, we estimate one of the phase functions.

3.2 Pseudo Period Estimation

First, we define a differential of the phase function

$$\frac{ds_Q(t)}{dt} = \frac{1}{P_Q(t)}, \tag{6}$$

where $P_Q(t)$ is called the *pseudo period* in this paper. Note that the pseudo period $P_Q(t)$ is equivalent to the period P for the periodic signal, which is obvious from Eq. (2). The equation in quantized domain is also defined as

$$s_{Q,i+1} - s_{Q,i} = \frac{1}{P_{Q,i}}. \tag{7}$$

Then, the pseudo period is estimated by maximizing a short-term normalized autocorrelation as

$$\hat{P}_{Q,i} = \arg \max_{P_Q \in [P_{min}, P_{max}]} C_i(P_Q) \tag{8}$$

$$C_i(P_Q) = \frac{\sum_{\tau \in I_i} \boldsymbol{g}_\tau^T \boldsymbol{g}_{\tau+P_Q}}{\sqrt{\sum_{\tau \in I_i} ||\boldsymbol{g}_\tau||^2} \sqrt{\sum_{\tau \in I_i} ||\boldsymbol{g}_{\tau+P_Q}||^2}} \tag{9}$$

$$I_i = \{\tau \mid i - \alpha P_{max} \leq \tau \leq i + \alpha P_{max}, \tau \in \mathbb{Z}\}, \tag{10}$$

where $[P_{min}, P_{max}]$ is a domain of the pseudo period which is obtained by existing methods of period detection or given by prior knowledge, and α is a coefficient to control the size of the window function for the short-term mask.

3.3 Self Dynamic Time Warping

Given a correspondence of two samples i and u_i^j with j periods difference (call this the jth period correspondence and denote it as $\boldsymbol{x} = [i, u]$), they are ideally subject to

$$s_{Q,u_i^j} - s_{Q,i} = j \tag{11}$$

$$\boldsymbol{g}_{u_i^j} = \boldsymbol{g}_{i\cdot}, \tag{12}$$

where the equations denote the phase constraint and signal consistency, respectively. Hence, Eq. (11) is exploited as a constraint for phase registration, and we try to find the correspondences based on the signal consistency of Eq. (12) by applying Self Dynamic Time Warping (Self DTW) to the quasi periodic sequence $\boldsymbol{G}$.

First, an initial estimate of the jth period correspondence $\hat{\boldsymbol{x}}_i^j = [i, \hat{u}_i^j]$ is obtained from the estimated pseudo period $\hat{P}_{Q,i}$ in a recursive manner as

$$\hat{u}_i^j = \hat{u}_i^{j-1} + \hat{P}_{Q,\hat{u}_i^{j-1}},\ \hat{u}_i^0 = i \tag{13}$$

Second, lower and upper bounds of the jth period correspondence are set to

$$u_{low,i}^j = \max\{\hat{u}_i^j - \beta \hat{P}_{Q,\hat{u}_i^j}, 0\} \tag{14}$$

$$u_{up,i}^j = \min\{\hat{u}_i^j + \beta \hat{P}_{Q,\hat{u}_i^j}, N\}. \tag{15}$$

Thus, a Self DTW path search region is defined as $R^j = \{\boldsymbol{x} = [i, u] |\ \hat{u}_{low,i}^j \leq u \leq \hat{u}_{up,i}\ \forall i \in [0, N]\}$, and subsequently the source and terminal regions are set

to $R_S^j = \{\boldsymbol{x} = [0, u] | \ \boldsymbol{x} \in R^j\}$ and $R_T^j = \{\boldsymbol{x} = [i, N] | \ \boldsymbol{x} \in R^j\}$, respectively, as illustrated in Fig. 1. Now, the correspondence problem is decoded as continuous dynamic programming [6] in the search region R^j.

The formulation is given as follows. A cumulative cost $c(\boldsymbol{x})$ and a counter $n(\boldsymbol{x})$ are introduced and these are initialized for $\boldsymbol{x} \in R_S^j$ as

$$c(\boldsymbol{x}) = c_I(\boldsymbol{x}), \ n(\boldsymbol{x}) = 1, \tag{16}$$

where $c_I(\boldsymbol{x})$ is a cost function for the signal intensity difference given as $c_I(\boldsymbol{x}) = ||\boldsymbol{g}_i - \boldsymbol{g}_u||$.

Next, a transition process is considered. We limit the previous state $\boldsymbol{x}_p$ to the current state $\boldsymbol{x}$ to $T^j(\boldsymbol{x}) = \{[i-1, u-1], [i-2, u-1], [i-1, u-2]\} \cap R^j$ and define the optimal previous state to the current state as $\boldsymbol{x}_p^{j\,*}(\boldsymbol{x})$, which is given as

$$\boldsymbol{x}_p^{j\,*}(\boldsymbol{x}) = \arg \min_{\boldsymbol{x}_p \in T^j(\boldsymbol{x})} \left\{ \frac{c(\boldsymbol{x}_p)}{n(\boldsymbol{x}_p)} + c_T(\boldsymbol{x}, \boldsymbol{x}_p) \right\}, \tag{17}$$

where the first and second terms on the right side are, respectively, the counter-normalized previous cumulative cost and the transition cost function, given as $c_T(\boldsymbol{x}, \boldsymbol{x}_p) = ||\boldsymbol{x} - \boldsymbol{x}_p||_{L_1}$. Then, the cumulative cost and the counter are updated as

$$c(\boldsymbol{x}) = c(\boldsymbol{x}_p^j(\boldsymbol{x})^*) + c_I(\boldsymbol{x}) + c_T(\boldsymbol{x}, \boldsymbol{x}_p^{j\,*}(\boldsymbol{x})) \tag{18}$$

$$n(\boldsymbol{x}) = n(\boldsymbol{x}_p^{j\,*}(\boldsymbol{x})) + 1 \tag{19}$$

After the cost propagation of all the states in R^j, the optimal state at the terminal is

$$\boldsymbol{x}_T^{j\,*} = \arg \min_{\boldsymbol{x} \in R_T^j} \frac{c(\boldsymbol{x})}{n(\boldsymbol{x})}. \tag{20}$$

Subsequently, the terminal counter and the optimal terminal state are redefined, respectively, as $n^j = n(\boldsymbol{x}_T^{j\,*})$ and $\boldsymbol{x}_{n_j}^{j\,*} = \boldsymbol{x}_T^{j\,*}$ for convenience, and the optimal path is back tracked as $\boldsymbol{x}_i^{j\,*} = \boldsymbol{x}_p^{j\,*}(\boldsymbol{x}_{i+1}^{j\,*})$ for $i = n^j - 1, \ldots, 1$. In the following sections, the optimal correspondence sequence is denoted as $\boldsymbol{X}^j = \{\boldsymbol{x}_i^j | \ i = 1, \ldots, n^j\}$.

3.4 Phase Sequence Optimization

Phase sequence $\boldsymbol{S}_Q$ is estimated by taking the following three points into consideration: (1) the obtained optimal correspondence sequence $\boldsymbol{X}^j$, (2) the smoothness of the phase sequence $\boldsymbol{S}_Q$, and (3) monotonically increasing the phase sequence $\boldsymbol{S}_Q$ as

$$\boldsymbol{S}_Q^* = \arg \min_{\boldsymbol{S}_Q} D(\boldsymbol{S}_Q) \tag{21}$$

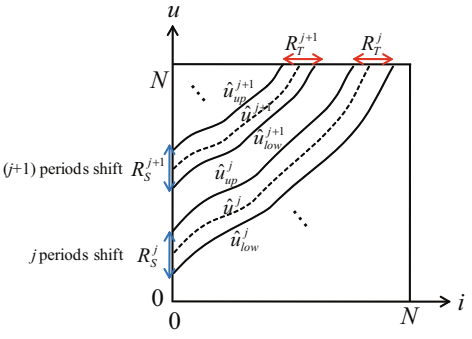


Fig. 1. Overview of Self DTW

$$D(\boldsymbol{S}_Q) = \sum_j \sum_{[i,u]\in \boldsymbol{X}^j} (s_{Q,u} - s_{Q,i} - j)^2 + \lambda \sum_{i=0}^{N-1} \left(s_{Q,i+1} - s_{Q,i} - \frac{1}{\hat{P}_{Q,i}} \right)^2 \tag{22}$$

$$\text{subject to } s_{Q,i+1} - s_{Q,i} \geq 0 \ \forall i = 0, \ldots, N-1, \tag{23}$$

where the first and second terms on the right side of Eq. (22) are the data term derived from the correspondences and the regularization term derived from Eqs. (7) and (8), respectively, and λ is the regularization term coefficient.

As described before, the ambiguity of the phase function remains. First, a constant shift Δs in $s'_Q(t) = s_Q(t) + \Delta s$ does not change the value of the objective functions and the constraints at all, because all the $s_{Q,i}$ are used in the subtraction form. Therefore, the following constraint is added without loss of generality

$$s_{Q,0} = 0. \tag{24}$$

Second, considering another phase function $s'_Q(t) = s_Q(t) + r(t)$ with a quasi-periodic shift $r(t)$ that satisfies

$$r(t) = r(t') \ \forall [t, t'] \in \{[t, t'] |\ \tilde{s}_Q(t') = \tilde{s}_Q(t)\} \tag{25}$$

$$\frac{dr(t)}{dt} \geq -\frac{ds_Q(t)}{dt}, \tag{26}$$

the quasi-periodic shift $r(t)$ does not change the data term of the objective functions assuming no correspondence error. In other words, the quasi-periodic shift $r(t)$ depends on a tradeoff between the correspondence errors in the data term and residuals between inverses of the correct pseudo period $P_{Q,i}$ and its estimate $\hat{P}_{Q,i}$ in the regularization term.

Finally, because the objective function $D(\boldsymbol{S}_Q)$ is a quadratic form and the constraints of Eqs. (23) and (24) are a linear form, the above optimization problem is solved by convex quadratic programming using the active set method.

4 Experiments

4.1 Simulation Data

We carried out experiments on simulation data to confirm the effectiveness of the proposed phase registration. First, we generated three normalized periodic functions with a single dimension as a non-parametric function, with the second order differential (d^2h/ds^2) randomly drawn from a uniform distribution in the domain [-500,500] and with boundary conditions $h(1) = h(0) = 0$. The phase function $s_Q(t)$ was also generated by a non-parametric scheme in the same way. Given the pseudo period function $P_Q(t)$ with second order differential (d^2P_Q/dt^2) drawn from a uniform distribution in the domain [-0.25, 0.25] with boundary conditions $P_Q(0) = P_Q(T) = P$, where T is the time at the final frame and P is a predefined period, the phase function $s_Q(t)$ is given by the first order differential equation $ds_Q/dt = 1/P_Q(t)$ with initial condition $s_Q(t) = 0$. In this simulation, T and P were set to 10 and 100, respectively.

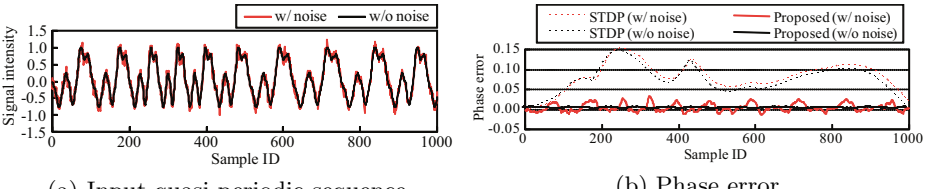
Third, quasi-periodic sequences were generated by sampling at $(1/P)$ intervals as $g_i = h(s_Q(it/P)),\ i = 0, \ldots, N$, where $N = TP$ is the sample ID at the final frames. Fourth, sequences with noise were also generated as $g'_i = g_i + \delta$, where δ is drawn from a Gaussian distribution with standard deviation $\sigma = 0.1$. The other parameters used in each process were set experimentally as $\alpha = 1.0$, $\beta = 0.3$, and $\lambda = 10.0$. The generated signals were phase-modulated as shown in Fig. 2(a).

If a reference signal is not given in the problem statement, existing methods such as continuous cyclic DP and cyclic HMM cannot be applied. Therefore, we regard the following scheme based on the estimated pseudo period with Short-Term Period Detection (STPD) as a baseline algorithm for comparison:

$$s_{Q,i+1} = s_{Q,i} + \hat{P}_{Q,i}, \tag{27}$$

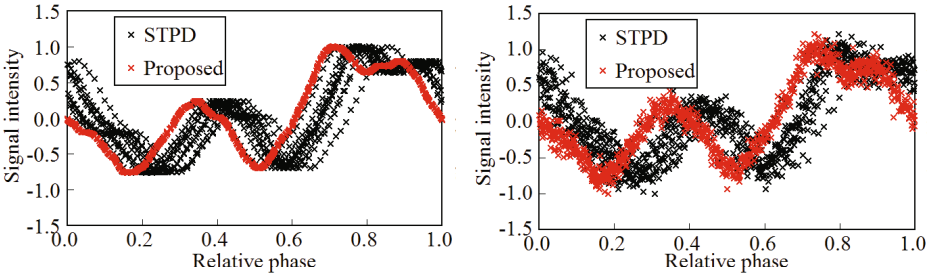
where we initialize $s_{Q,0} = 0$. Note that this is also equivalent to the case in which the regularization coefficient λ is set to infinity in the proposed framework.

First, we evaluated the errors between the estimated phase and the ground truth in Fig. 2(b). Because the ambiguity of the phase function is as described previously, bias components in the errors should be ignored here. As a result, the error variance in noisy data is larger than that in data without noise in the proposed method. The error patterns are, however, still similar to a quasi-periodic form; this implies the possibility of another combination of the phase function $s'_Q(t)$ and the



(a) Input quasi-periodic sequence (b) Phase error

Fig. 2. Quasi-periodic input sequence and its phase error



(a) Without noise (b) With noise

Fig. 3. Phase registration results for simulated data

normalized periodic function $h'(s)$. On the other hand, the error variance in the baseline method (STPD) is larger than that in the proposed method, and furthermore, the error patterns are not similar to a quasi-periodic form.

Next, phase registration results were evaluated in the domain of the relative phase $\tilde{s}^*_{Q,i}$ and the corresponding signal intensity g_i in Fig. 3. Note that their plots form a certain normalized periodic function $h(s)$ if the phase is correctly registered. As a result, the plots for the data without noise in the proposed method lie on a single curve and they form similar curves to the original signals (Fig. 3(a)). Moreover, the plots for the noisy data seem to lie within the range of additive noise distribution in the quasi-periodic sequence (Fig. 3(b)). On the other hand, the plots of the baseline method are widely distributed around the original signals due to incorrect phase registration.

4.2 Real Data

We also conducted an experiment on a gait silhouette sequence with gradual speed variations ranging from 6 km/h to 10 km/h as shown in Fig. 4. An image sequence of a walking person on a speed-controllable treadmill was captured at 60 fps and a size-normalized silhouette sequence (88 by 128 pixels) extracted by graph cut-based segmentation in conjunction with background subtraction [26]. PCA was then applied to the silhouette images and subsequently, the proposed method was applied to the dimension-reduced data.

Figure 5 shows gait silhouette images aligned at the estimated relative phase. Despite the significant variation in gait style due to large speed variations from walking (6 km/h) to running (10 km/h), all the gait phases, such as double-support phase and single-support phase, are well registered for the different speeds. Note that non-uniform alignment intervals of the gait silhouette images in Fig. 5 represent non-linear time distortion due to the gait fluctuation obtained by the proposed Self DTW.

From an application viewpoint, phase-registered image sequences are quite useful. For example, given just a single walking sequence with speed variation, a gait manifold parameterized by both phase and walking speed can be constructed by re-sampling the phase-registered speed-varied gait image sequence as shown in

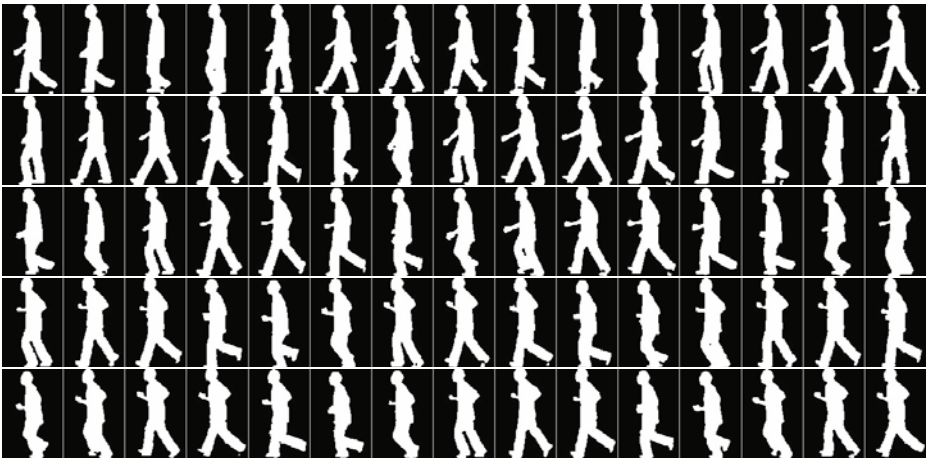


Fig. 4. Subsequences of input gait silhouettes (every 4 frames). Top to bottom rows correspond to 6, 7, 8, 9, and 10 km/h, respectively. Note that the phases among different walking speeds are not synchronized.

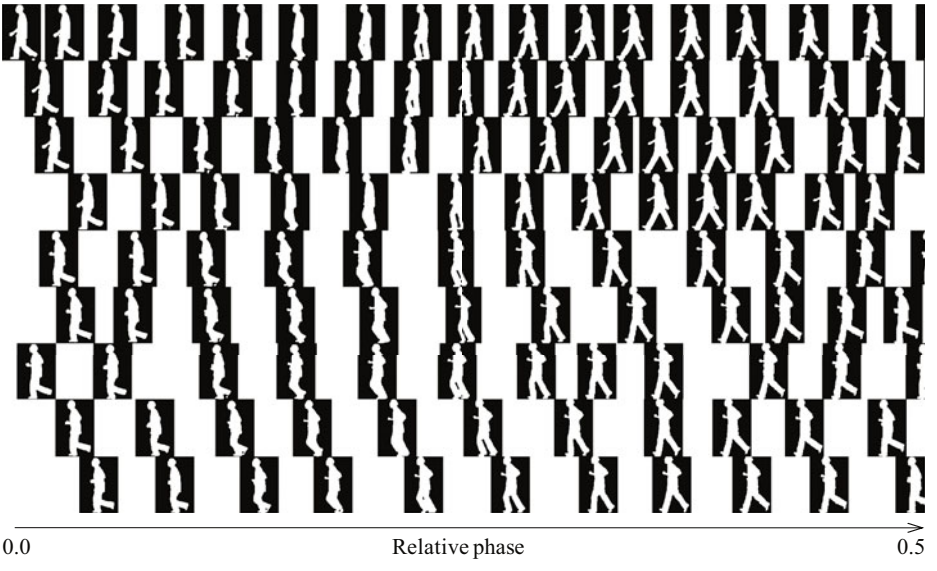


Fig. 5. Gait silhouette images aligned at the estimated phases (every 2 frames, a half gait period). The horizontal axis indicates the relative phase $\tilde{s}$ and each silhouette image is aligned at the estimated relative phase. The vertical axis indicates the number of periods (every 5 periods). Changes in the rows from top to bottom represent a gradual speed increase from 6 km/h to 10 km/h.

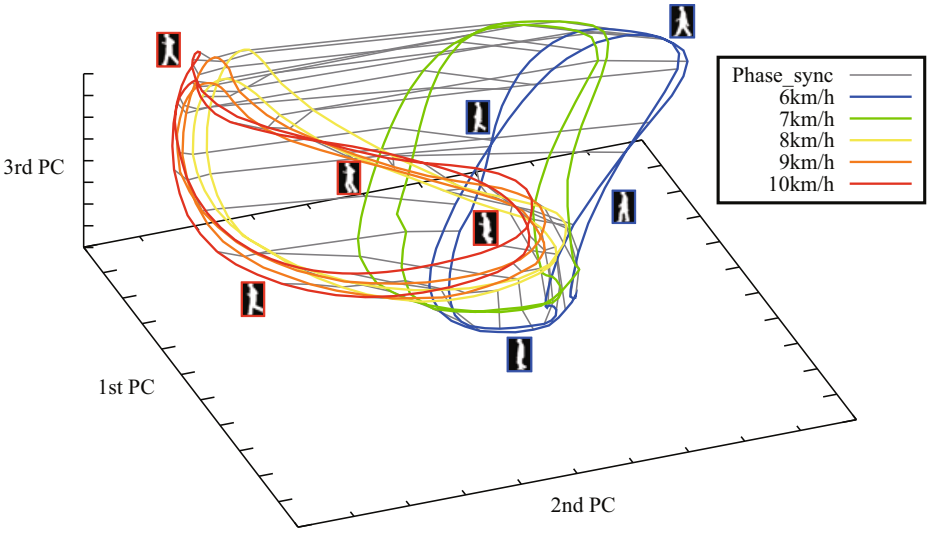


Fig. 6. A 2D gait manifold parameterized by phase and walking speed. While each color loop depicts a manifold for each walking speed parameterized by phase, gray lines represent phase synchronization among the walking speeds.

Fig. 6. The gait manifold enables us to analyze the gait pose transition by walking speed for the same phase as well as that by phase for the same walking speed. Moreover, in the context of gait recognition with speed variations, the 2D gait manifold is provided as an efficient gallery expression, unlike the existing 1D gait manifold parameterized only by phase [16]. A set of 1D gait manifolds with different speeds depicted as colored loops in Fig. 6 cannot deal with variations in walking speed within a period, particularly as they do not provide phase registration information for different walking speeds depicted as gray lines in Fig. 6. On the other hand, since a 2D gait manifold has such phase registration information for different walking speeds, it can appropriately match a sequence with walking speed variations within a period in the framework of 1D-2D (input to gallery) dynamic programming. Note that the proposed method is applicable not only to gait with speed variation, but also to general quasi-periodic signals undergoing transition by factors other than phase, such as periodic action recognition with gradual view changes or periodic signal analysis with gradual attenuation[1].

5 Conclusion

This paper proposed a method for phase registration of a single non-parametric quasi-periodic signal. Having detected a short-term period for each sample by

[1] In these cases, the manifold is parameterized by phase and view or degree of attenuation.

normalized autocorrelation, correspondences of multiple-period shifts are obtained by Self Dynamic Time Warping (Self DTW), which are used in the subsequent phase optimization framework.

Future works include eliminating ambiguity between the phase function and normalized period function based on the periodicity of the estimated phase sequence, extension of the proposed method for a quasi-periodic signal with both phase and amplitude modulation, and application to matching and time super-resolution of the quasi-periodic signals.

Acknowledgement. This work was supported by Grant-in-Aid for Scientific Research(S) 21220003.

References

1. Newkirk, D., Karlquist, R.: Communication systems, 2nd edn. McGraw-Hill, Inc., New York (1981)
2. Anderson, J.B., Aulin, T., Sundberg, C.E.: Digital phase modulation. Springer, Heidelberg (1986)
3. Sarkar, S., Phillips, J., Liu, Z., Vega, I., Grother, P., Bowyer, K.: The humanid gait challenge problem: Data sets, performance, and analysis. Trans. of Pattern Analysis and Machine Intelligence 27, 162–177 (2005)
4. van Ouwerkerk, J.: Image super-resolution survey. Image and Vision Computing 24, 1039–1052 (2006)
5. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on Acoustics, Speech and Signal Processing 26, 43–49 (1978)
6. Oka, R.: Spotting method for classification of real world data. Computer Journal 41, 559–565 (1998)
7. Aronsson, D., Bjornemo, E., Johansson, M.: Estimation and detection of a periodic signal. In: Proc. of American Institute of Physics Conference, Bayesian Inference and Maximum Entropy Methods In Science and Engineering, vol. 872, pp. 139–146 (2006)
8. Znak, V.: Some aspects of estimating the detection rate of a periodic signal in noisy data and the time position of its components. Pattern Recognition and Image Analysis 19, 539–545 (2009)
9. Handel, P., Tichavsky, P.: Adaptive estimation for periodic signal enhancement and tracking. International Journal of Adaptive Control and Signal Processing 8, 447–456 (2007)
10. Barros, A.K., Ohnishi, N.: Amplitude estimation of quasi-periodic physiological signals by wavelets. IEICE Transactions on Information and Systems E83-D, 2193–2195 (2000)
11. Gruber, P., Todtli, J.: Estimation of quasiperiodic signal parameters by means of dynamicsignal models. IEEE Transactions on Signal Processing 42, 552–562 (1994)
12. Nakashizuka, M.: A sparse decomposition method for periodic signal mixtures. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences E91-A, 791–800 (2008)
13. Wong, H., Sethares, W.A.: Estimation of pseudo-periodic signals. In: Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, vol. 2, pp. 557–560 (2004)

14. Murase, H., Sakai, R.: Moving object recognition in eigenspace representation: Gait analysis and lip reading. Pattern Recognition Letters 17, 155–162 (1996)
15. Boulgouris, N., Plataniotis, K., Hatzinakos, D.: Gait recognition using linear time normalization. Pattern Recognition 39, 969–979 (2006)
16. Mori, A., Makihara, Y., Yagi, Y.: Gait recognition using period-based phase synchronization for low frame-rate videos. In: Proc. of 20th Int. Conf. on Pattern Recognition, Istanbul, Turkey (2010)
17. Little, J., Boyd, J.: Recognizing people by their gait: The shape of motion. Videre: Journal of Computer Vision Research 1, 1–13 (1998)
18. Makihara, Y., Sagawa, R., Mukaigawa, Y., Echigo, T., Yagi, Y.: Gait recognition using a view transformation model in the frequency domain. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3953, pp. 151–163. Springer, Heidelberg (2006)
19. Veeraraghavan, A., Chellappa, R., Roy-Chowdhury, A.: The function space of an activity. In: Proc. of the 2006 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, New York, USA, vol. 1, pp. 959–966 (2006)
20. Cuntoor, N., Kale, A., Chellappa, R.: Combining multiple evidences for gait recognition. In: Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, vol. 3, pp. 33–36 (2003)
21. Zhou, Z., Damper, R., Prugel-Bennett, A.: Model selection within a bayesian approach to extraction of walker motion. In: Proc. of the IEEE Computer Society Workshop on Biometrics 2006, New York, USA (2006)
22. Lee, L., Dalley, G., Tieu, K.: Learning pedestrian models for silhouette refinement. In: Proc. Int'l Conf. on Computer Vision 2003, vol. 1, pp. 663–670 (2003)
23. Liu, Z., Sarkar, S.: Effect of silhouette quality on hard problems in gait recognition. Trans. of Systems, Man, and Cybernetics Part B: Cybernetics 35, 170–183 (2005)
24. Liu, Z., Sarkar, S.: Improved gait recognition by gait dynamics normalization. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 863–876 (2006)
25. Sunderesan, A., Chowdhury, A., Chellappa, R.: A hidden markov model based framework for recognition of humans from gait sequences. In: IEEE Int'l Conf. on Image Processing 2003, vol. 2, pp. 93–96 (2003)
26. Makihara, Y., Yagi, Y.: Silhouette extraction based on iterative spatio-temporal local color transformation and graph-cut segmentation. In: Proc. of the 19th Int. Conf. on Pattern Recognition, Tampa, USA, Florida (2008)

Latent Gaussian Mixture Regression for Human Pose Estimation

Yan Tian[1,3], Leonid Sigal[2], Hernán Badino[3], Fernando De la Torre[3], and Yong Liu[1]

[1] Beijing University of Posts and Telecommunications, Beijing, P.R. China
[2] Disney Research, Pittsburgh, US
[3] Carnegie Mellon University, Pittsburgh, US

Abstract. Discriminative approaches for human pose estimation model the functional mapping, or conditional distribution, between image features and 3D pose. Learning such multi-modal models in high dimensional spaces, however, is challenging with limited training data; often resulting in over-fitting and poor generalization. To address these issues latent variable models (LVMs) have been introduced. Shared LVMs attempt to learn a coherent, typically non-linear, latent space shared by image features and 3D poses, distribution of data in that latent space, and conditional distributions to and from this latent space to carry out inference. Discovering the shared manifold structure can, in itself, however, be challenging. In addition, shared LVMs models are most often non-parametric, requiring the model representation to be a function of the training set size. We present a parametric framework that addresses these shortcoming. In particular, we learn latent spaces, and distributions within them, for image features and 3D poses separately first, and then learn a multi-modal conditional density between these two low-dimensional spaces in the form of Gaussian Mixture Regression. Using our model we can address the issue of over-fitting and generalization, since the data is denser in the learned latent space, as well as avoid the necessity of learning a shared manifold for the data. We quantitatively evaluate and compare the performance of the proposed method to several state-of-the-art alternatives, and show that our method gives a competitive performance.

1 Introduction

Monocular pose estimation has been a focus of much research in vision due to abundance of applications for marker-less motion capture in activity recognition and human computer interaction. Despite much research, however, monocular pose estimation remains a difficult task; challenges include high-dimensionality of the state space, image clutter, occlusions, lighting and appearance variations, to name a few.

Most prior works can be classified into two classes of approaches: *generative* and *discriminative*. *Generative* approaches [1,2] define an image formation model by predicting appearance of the body $\mathbf{x}$ given a hypothesized state of

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 679–690, 2011.

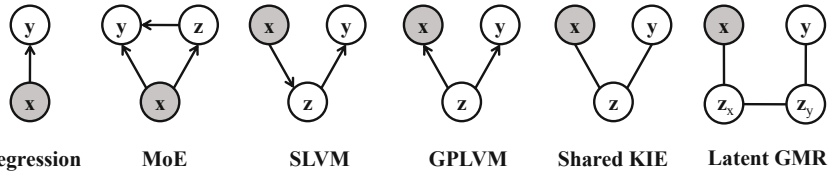


Fig. 1. Graphical model representations of models used for discriminative human pose estimation, including Regression Models [3,13], Mixture Models (*e.g.*, Mixture of Experts (MoE) [4,14]), Spectral Latent Variable Models (SLVM) [11], Gaussian Process Latent Variable Models [17,12] and Shared Kernel Information Embeddings (sKIE) [18]. In all illustrations $\mathbf{x}$ denotes observed input variable corresponding to image features, $\mathbf{y}$ denotes the inferred 3D pose, and $\mathbf{z}$ corresponds to auxiliary latent variables (in case of Mixture of Gaussians (MoE) corresponding to the latent mixture component identity).

the body (pose) $\mathbf{y}$; an inference framework is then used to infer the posterior, $p(\mathbf{y}|\mathbf{x}) \propto p(\mathbf{x}|\mathbf{y})p(\mathbf{y})$ over time. Since the inference often takes the form of non-convex search in a high-dimensional space of body articulations, these methods are computationally expensive and can suffer from local convergence (typically requiring a good initial guess for pose to seed the search).

Discriminative approaches [3,4,5,6,7,8,9,10,11,12,13,14,15,16] avoid building an explicit imaging model, and instead opt to learn regression function, $\mathbf{y} = f(\mathbf{x})$, that maps from image features, $\mathbf{x}$, to 3D pose, $\mathbf{y}$; or probabilistically, a conditional distribution $p(\mathbf{y}|\mathbf{x})$ directly. The main goal is to learn a model from labeled training data, $\{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^{N}$, that provides efficient and effective generalization for new examples at test time. The difficulty with this class of methods is twofold: (1) the conditional probability of pose given image features, $p(\mathbf{y}|\mathbf{x})$, is typically multi-modal: different image features can be explained by several poses; and (2) learning high dimensional regression functions, or conditional distributions, using limited training data is challenging and often results in over-fitting. Here we focus on discriminative pose estimation.

Discriminative methods can further be categorized into: *parametric* and *non-parametric*. *Parametric* methods are appealing because the model representation is fixed[1]. Simple parametric models, *e.g.*, Linear Regression (LR) [3] or Relevance Vector Machine (RVM) [3,10], however, are (i) unable to deal with a multi-modal nature of the problem and (ii) unable to model the fine non-linear relationship between image features and pose. *Non-parametric* methods, *e.g.*, Nearest Neighbor Regression [13] or Kernel Regression [13], are able to model arbitrary complex relationships between input features and output poses, subject to the availability of the training data.

To deal with multi-modality, on the parametric side, mixture models were introduced, *e.g.*, Mixture of Regressors [4] or Mixture of Experts [14]. On the non-parametric side, local models that cluster data into convex sets and use uni-modal predictions within each cluster became popular (*e.g.*, Local Gaussian

[1] Complexity of the model is not a function of the number of training examples.

Process Latent Variable Models (Local GPLVM) [16]). In both cases over-fitting and generalization remained an issue, due to the need for large training datasets, as noted in [12].

To alleviate this problem Latent Variable Models (LVMs) were introduced as an intermediate representation. Kanaujia *et al.*, [11], proposed Spectral LVMs to learn a non-linear latent embedding of the 3D pose data and a separately trained mixture model to map from the image features to the plausible latent positions in the sub-space. The relationship between the image features and latent space, however, was assumed to be linear within each mixture component. As an alternative, Shared Gaussian Processes Latent Variable Model (Shared GPLVM) was introduced in [12] and [17], where the latent embedding was learned to preserve the joint structure of image features and 3D poses simultaneously; the forward non-linear mappings from the latent space to the input and output spaces were also learned at the same time. Due to the lack of backward mapping from the image features to the latent space, inference remained expensive, requiring multiple optimizations at the cost of $O(N^2)$, where N is the number of training examples. Shared Kernel Information Embeddings (sKIE) [18] provided closed form mappings to and from the latent space reducing the training and inference complexity by an order of magnitude. Both Shared GPLVM and sKIE are non-parametric, with the model complexity being a function of the training set size; this makes them less appealing for use with larger datasets.

We present a parametric counterpart framework to the non-parametric latent models discussed above.

1. We learn a multi-modal joint density model between the image features and the 3D pose, in the form of a Gaussian Mixture Model (GMM). GMM allows us to deal with multi-modality in the data and derive explicit conditional distributions for inference, in the form of Gaussian Mixture Regression (GMR).
2. To alleviate the need for large training sets while at the same time limiting over-fitting, we formulate the GMM learning in the latent spaces for both image features and 3D pose.
3. Since the manifold structure of both image features and 3D poses is complex and cannot be well approximated by simple linear latent spaces, we propose to use Locality Preserving Projections (LPP) [19] that while learning linear mapping can discover non-linear manifold structure [19]. LPP also provides us with closed form forward and backward mappings between the latent space(s) and input/output space(s).

As a result our model is able to: (1) deal with multi-modalities in the data, (2) model complex structure of the image feature and pose manifolds, (3) provides both forward and backwards mapping between the respective manifolds and original image feature or pose spaces, and (4) alleviates the need for learning, a sometimes hard to obtain[2], shared manifold structure.

[2] Shared manifold structure can be hard to obtain, for example, if the input and output features have vastly different dimensionality.

2 Gaussian Mixture Regression

Non-parametric regression methods rely on manifold local smoothness in a typically high-dimensional input/output spaces to model the regression function; however, they can suffer from local sparsity problems. When the data is sparse (which is typically the case for high-dimensional spaces) and a test point is far from the training data, the kernels tend to produce poor estimates. In addition, the complexity of non-parametric methods is typically a function of the training set size (*e.g.*, $O(N)$ for KIE and $O(N^2)$ for GPLVM), making them hard to scale to large datasets. In this paper, we employ a parametric Gaussian Mixture Regression to address these problems.

Given observations (*e.g.*, image features), $\mathbf{x} \in \mathbb{R}^{d_\mathbf{x}}$, and targets (*e.g.*, 3D poses), $\mathbf{y} \in \mathbb{R}^{d_\mathbf{y}}$, where $d_\mathbf{x}$ is dimensionality of the observation, and $d_\mathbf{y}$ is dimensionality of the target space, we assume the joint data samples, $(\mathbf{x}, \mathbf{y})$, follow the Gaussian mixture distribution with K mixture components,

$$P(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^{K} \pi_k P(\mathbf{x}, \mathbf{y}; \mu_k, \mathbf{\Lambda}_k) \tag{1}$$

where $P(\mathbf{x}, \mathbf{y}; \mu_k, \mathbf{\Lambda}_k)$ is the multivariate Gaussian density function. The parameters of model include prior weights, π_k, means, $\mu_k = [\mu_{k,\mathbf{x}}\ \mu_{k,\mathbf{y}}]^T$, and variances, $\mathbf{\Lambda}_k = [\mathbf{\Lambda}_{k,\mathbf{x}}\ \mathbf{\Lambda}_{k,\mathbf{xy}}; \mathbf{\Lambda}_{k,\mathbf{yx}}\ \mathbf{\Lambda}_{k,\mathbf{y}}]$, of each Gaussian component.

The joint density can be expressed as the sum of the products of the marginal density of $\mathbf{x}$, and the probability density function of $\mathbf{y}$ conditioned on $\mathbf{x}$:

$$P(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^{K} \pi_k P(\mathbf{y}|\mathbf{x}; m_k, \sigma_k^2) P(\mathbf{x}; \mu_{k,\mathbf{x}}, \mathbf{\Lambda}_{k,\mathbf{x}}). \tag{2}$$

Similarly, the marginal distribution,

$$P(\mathbf{x}) = \sum_{\mathbf{y}} P(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^{K} \pi_k P(\mathbf{x}; \mu_{k,\mathbf{x}}, \mathbf{\Lambda}_{k,\mathbf{x}}), \tag{3}$$

is also a mixture.

The global regression function can be obtained by combing (2) and (3):

$$P(\mathbf{y}|\mathbf{x}) = \frac{P(\mathbf{x}, \mathbf{y})}{P(\mathbf{x})} = \frac{\sum_{k=1}^{K} \pi_k P(\mathbf{x}; \mu_{k,\mathbf{x}}, \mathbf{\Lambda}_{k,\mathbf{x}}) P(\mathbf{y}|\mathbf{x}; m_k, \sigma_k^2)}{\sum_{k=1}^{K} \pi_k P(\mathbf{x}; \mu_{k,\mathbf{x}}, \mathbf{\Lambda}_{k,\mathbf{x}})} \tag{4}$$

This can be expressed as a mixture of conditional distributions, $P(\mathbf{y}|\mathbf{x}) = \sum_{k=1}^{K} \omega_k P(\mathbf{y}|\mathbf{x}; m_k, \sigma_k^2)$, where the mixing weights ω_k are defined as:

$$\omega_k = \frac{\pi_k P(\mathbf{x}; \mu_{k,\mathbf{x}}, \mathbf{\Lambda}_{k,\mathbf{x}})}{\sum_{j=1}^{K} \pi_j P(\mathbf{x}; \mu_{j,\mathbf{x}}, \mathbf{\Lambda}_{j,\mathbf{x}})}. \tag{5}$$

The mean and the variance of the conditional distribution $P(\mathbf{y}|\mathbf{x})$ can be acquired in closed form by:

$$m_k = \mu_{k,\mathbf{x}} + \mathbf{\Lambda}_{k,\mathbf{yx}}\mathbf{\Lambda}_{k,\mathbf{x}}^{-1}(\mathbf{x} - \mu_{k,\mathbf{x}}) \tag{6}$$

$$\sigma_k^2 = \mathbf{\Lambda}_{k,\mathbf{y}} - \mathbf{\Lambda}_{k,\mathbf{yx}}\mathbf{\Lambda}_{k,\mathbf{x}}^{-1}\mathbf{\Lambda}_{k,\mathbf{xy}} \tag{7}$$

The learning can be achieved with a simple Gaussian Mixture Model, using Expectation Maximization (EM) procedure with K-means initialization. The prediction given a new input can be obtained by computing expectation over $P(\mathbf{y}|\mathbf{x})$:

$$E[P(\mathbf{y}|\mathbf{x})] = \sum_{k=1}^{K} \omega_k m_k. \tag{8}$$

Alternatively, if the conditional relationship is truly multi-modal, it is better to look at the modes given by m_j directly. In general, we can have up to K distinct modes in the conditional distribution for a given input, $\mathbf{x}$.

Relationship to Other Methods. Notice that the regression function (8) derived from the joint mixture Gaussian density is of the form of a kernel estimator. However, there is a key difference with non-parametric regression: the mixture weights, ω_k, are not determined by the local structure of the data, but rather by the components of a global Gaussian mixture model.

The Nadaraya-Watson kernel smoother [20] is a Gaussian Mixture Regression model with $K = N$ components, where N is the total number of training points. At the other end of the spectrum, $K = 1$ is approximately the classical linear regression model. Hence, the Gaussian Mixture Regression model can, in principal, represent a spectrum of regression models, ranging from the non-parametric kernel regression, where $K = N$, to the classical linear regression, $K = 1$.

Mixture Gaussian Regression is also closely related to the Mixture of Regression model and Mixture of Experts model (with a particular form of experts and gaits). For more discussion of this, see [21], Section 2.2.3.

3 Latent GMR Body Pose Estimation

As described in the previous section, we could use image features for inputs and 3D poses for targets and learn a GMR model in the original high-dimensional space. This has two shortcomings, however: (1) this would involve estimation of large number of parameters and hence require lots of training data, and (2) this assumes essentially a piece-wise multi-linear relationship between image features and 3D pose. For these reasons, we postulate that learning GMR in the latent space of both, image features and pose, actually results in better generalization and overall quality of the model.

To test this assumption we run a simple illustrative experiment with canonical correlation analysis. Canonical correlation analysis (CCA) [22] is a technique to extract common features from a pair of multivariate data. CCA, first proposed by Hotelling in 1936 [23], finds linear basis vectors for two sets of variables, such

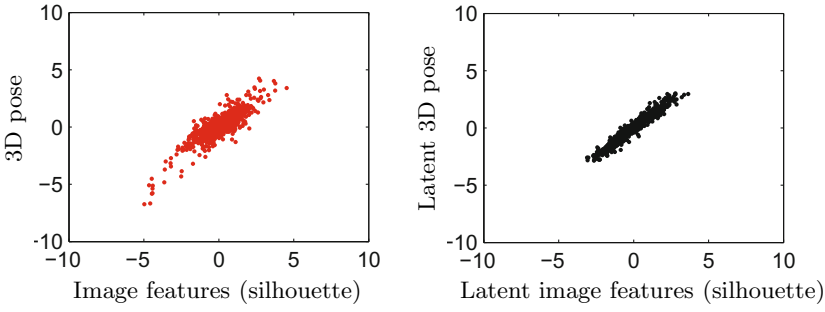


Fig. 2. Canonical component analysis of the silhouette and the human pose in the original and latent spaces (see text for more details)

that the correlation between the projections of variables onto these basis vectors are mutually maximized. We learn two CCA models based on 200 image-pose pairs: one for raw silhouette binary features $\in \mathbb{R}^{2450}$ and pose features encoded using 3D joint positions $\in \mathbb{R}^{69}$ (Fig. 2 (left)); and one for latent projections of the image and poses into 100 and 7 dimensional linear sub-spaces, obtained using PCA (Fig. 2 (right)); we illustrate only the first dimension along each of the axis in Fig. 2. It is clear from Fig. 2 that pose and image features are more closely correlated when projected into latent spaces (which reduces noise and optimally weights features). However, CCA is likely suffer from overfitting when having small training sets, and regularizing the solution introduces additional parameters to tune. Moreover, to model non-linear relations between image features and pose parameters kernel methods need to be applied, and it is unclear how to learn the functional form of the kernel and the kernel parameters specially in presence of limited training samples. In next section we propose to use Locality Preserving Projection (LPP) as an efficient and effective dimensionality reduction algorithm to capture the subtle manifold structure of the data. Additionally, LPP is not as prone to over-fitting and does not make assumptions about the global distribution of the data.

3.1 Locality Preserving Projections

Nonlinear dimensionality reduction techniques like Isomap [24], Locally Linear Embedding [10,7], or Gaussian Process Latent Variable Models [12] identify a low dimensional embedding of the data, but are defined only for the training data points (*i.e.*, only give a mapping from the manifold to the original data space); it is unclear how to obtain a latent position for a new test points. This makes inference challenging, often involving optimization [12] of the latent position based on the initial guess given by a set of nearest neighbors in the original space.

In contrast, the Locality Preserving Projections (LPP) [19], like PCA, can be simply applied to any new data point to locate it in the reduced representation space by finding the optimal linear approximations to the eigenfunctions of the

Laplace Beltrami operator on the manifold. Therefore, we use LPP to find low-dimensional embeddings of both image features and 3D poses.

For example, given a training dataset of N poses, $\mathbf{Y} = \{\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, ..., \mathbf{y}^{(N)}\} \in \mathbb{R}^{d_\mathbf{y} \times N}$, we want to find a transformation matrix $\mathbf{A} = [\mathbf{a}_1, ..., \mathbf{a}_{d_\mathbf{z}}]^T$ of basis vectors, $\mathbf{a}_i$, that maps these points to a set of latent points $\mathbf{Z} = \{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, ..., \mathbf{z}^{(N)}\} \in \mathbb{R}^{d_\mathbf{z} \times N}$ ($d_\mathbf{z} \ll d_\mathbf{y}$), such that $\mathbf{z}^{(i)}$ is a low dimensional manifold embedding representation of a high dimensional space pose $\mathbf{y}^{(i)}$. Following [19], this can be expressed as:

$$\begin{aligned} \min_{\mathbf{A}} \quad & \mathbf{A^T Y L Y^T A} \\ \text{subject to} \quad & \mathbf{A^T Y D Y^T A = I} \end{aligned} \tag{9}$$

Where $\mathbf{D}$ is a diagonal matrix whose entries are column sums of weight matrix $\mathbf{W}$, and $\mathbf{W}$ incurs a heavy penalty if neighboring training points are mapped far apart; $\mathbf{L = D - W}$ is Laplacian matrix.

3.2 Learning

Learning of the proposed model, is formulated as a three step procedure. Given a dataset of labeled feature-pose pairs, $\{\mathbf{x}^{(i)},\mathbf{y}^{(i)}\}_{i=1}^N$, we: (1) learn a low-dimensional embedding of the 3D pose data, $\{\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, ..., \mathbf{y}^{(N)}\}$, by solving optimization in Eq. 9; (2) learn a low-dimensional embedding of the image features by solving similar optimization for $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(N)}\}$; (3) learning a Gaussian Mixture Model (GMM) for the latent features and pose representations, $\{\mathbf{z}_\mathbf{x}^{(i)}, \mathbf{z}_\mathbf{y}^{(i)}\}_{i=1}^N$, obtained in (1) and (2).

3.3 Inference

Given a learned model, the inference for a new test image, represented in terms of image features $\hat{\mathbf{x}}$, involves: (1) getting a latent representation of $\hat{\mathbf{x}}$, $\hat{\mathbf{z}}_\mathbf{x}$, by applying a learned LPP mapping, $\mathbf{A}_\mathbf{x}$; (2) closed form conditioning of Gaussian Mixture Model (GMM), using $\hat{\mathbf{z}}_\mathbf{x}$, to obtain a Gaussian Mixture Regression (GMR) function; (3) inferring the latent 3D pose, $\hat{\mathbf{z}}_\mathbf{y}$, by either computing expectation over GMR (for uni-modal predictions) or using modes (for multi-modal predictions); (4) reconstructing the high-dimensional 3D pose from the latent estimate(s), by applying an inverse LPP mapping, $\mathbf{A}_\mathbf{y}$.

4 Experiments

4.1 Data Sets

We test the performance of our method on three datasets: (1) Poser dataset – synthetic sequences produced by Poser software [25], (2) CMU dataset – real image/mocap dataset publicly available from [26], and (3) standard dataset with provided error metrics made available by Agarwal and Triggs [3].

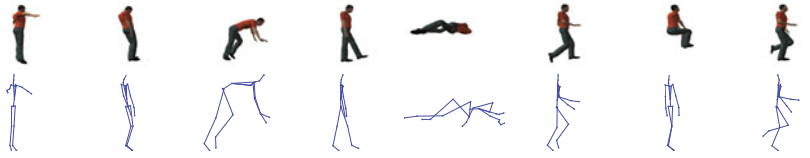


Fig. 3. Synthesized data generated by Poser 7 software

Poser dataset. We synthesize image data from motion capture sequences using Poser 7 software. The motion sequences come from 8 categories: walk, run, dance, fall, prone, sit, transitions and misc (see Fig. 3). A total of 5 sequences within each category are broken into: 3 training and 2 testing sequences, with each sequence containing approximately 500 frames. The size of each synthetic image is 500×490. We represent body pose in terms of 3D positions of 23 joints, resulting in $d_{\mathbf{y}} = 69$. All poses are represented in relative terms by subtracting the skeleton root (pelvis) from all other joint centers in every frame.

CMU dataset. From CMU Graphics Lab Motion Capture Database (see Fig. 4), we choose sequences of Subject 2 as training data, and use sequences in Subject

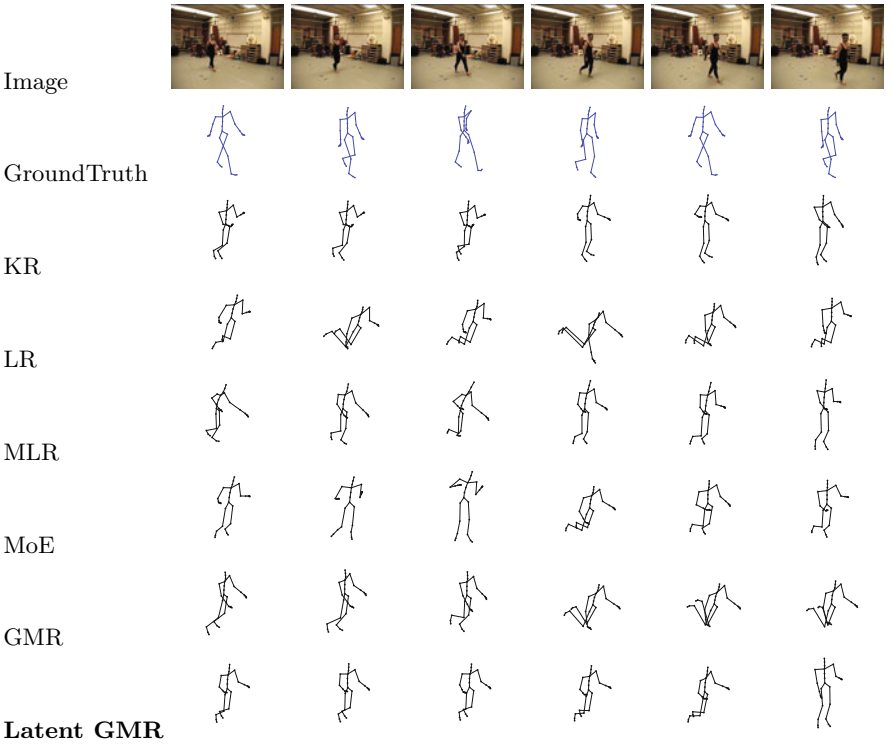


Fig. 4. Evaluation on frames: 38, 48, 58, 68, 78, and 88 of the 08-04 sequence from the CMU motion capture database

1, 8, 15 and 17 as test data. The size of each image is 240×352. We represent body pose in terms of 3D positions of 31 joints, resulting in $d_{\mathbf{y}} = 93$. Again, all poses are represented relative to the skeleton root (pelvis).

Image features. A number of representations for image features have been introduced over the years, *e.g.*, Scale invariant feature transform (SIFT) [15,6] or histogram of shape context [27,3,4,14], to name a few. Similar to prior work, we rely on silhouette features and encode them using a simpler 60D global shape context representation.

Error measure. We use a standard average joint position error inline as done by [18]. We report RMSE of average joint error in centimeters (cm).

Agarwal and Triggs dataset. To compare to other published techniques, we also utilize a publicly available benchmark dataset, that contains 1927 training and 418 test images, synthetically generated from mocap data. The pose is encoded using 54 joint angles in this case. The image features and error metric are provided with the dataset [3]. Silhouette features are represented using 100-dimensional feature vectors encoding the image silhouette using vector-quantized shape contexts. The mean RMSE error is computed over joint angles and is measured in degrees (for details see [3]).

4.2 Comparison

We compare our **Latent GMR** model with a number of alternatives, including: non-parametric regression model (kernel regression (**KR**)) and parametric regression models (linear regression (**LR**), mixture of linear regressors (**MLR**), mixture of experts (**MoE**), and Mixture Gaussian Regression (**MGR**)) in the originals high-dimensional space. The results are shown in Table 1 and Table 2. We use the same training and test datasets for all methods, and we also use a fixed set of parameters, for all sequences. For example, we train all the mixture models with $K = 8$ components. Other parameters are chosen by cross-validation: *e.g.*, the width of the RBF kernel in KR. In our method, the Locality Preserving Projections (LPP) is trained to keep 95% of the original energy. The results for [4,14] in Table 1 and Table 2 are based on re-implementations of the original work[3]. In all cases we compare the *expectations* computed under the models with ground truth.

We can see that since our features and data are sparse, kernel regression (KR) tends to work poorly in these cases. The performance of mixture models degrades as the data points start to fall close to the boundary between the two experts (since we are using expectation for inference). For this reason, sometimes the performance of mixture models is lower than that of uni-modal linear regression. Our Latent GRM model tends to produce better performance than competing methods.

Since the proposed Latent GMR contains two parts, *i.e.*, latent representation for the data and GMR model for inference, we attempt to study the interplay of

[3] For the purpose of comparison, we do not explore the temporal prior which is employed in [10].

Table 1. Evaluation of different algorithms on the Poser dataset (for details see text)

Error (cm)		KR	LR	MLR [4]	MoE [14]	GMR	Latent GMR
dance	S1	10.85	5.83	5.76	5.72	7.79	**5.60**
	S2	10.37	5.23	5.10	5.04	6.23	**4.91**
falls	S1	15.32	10.40	10.27	10.25	10.82	**10.05**
	S2	16.31	11.50	11.32	11.26	12.99	**10.92**
miscs	S1	8.32	3.59	3.53	3.42	3.86	**3.28**
	S2	19.27	12.19	12.11	12.10	14.44	**11.80**
prone	S1	11.36	6.55	6.46	6.40	7.06	**5.88**
	S2	12.46	6.36	6.32	6.28	7.00	**6.19**
run	S1	8.94	4.70	4.65	4.64	6.23	**4.31**
	S2	11.65	5.96	5.85	5.79	6.89	**5.62**
sit	S1	18.56	13.29	13.24	13.20	14.14	**13.01**
	S2	9.65	4.92	4.87	4.81	6.03	**4.24**
transition	S1	11.23	5.78	5.50	5.44	6.17	**5.32**
	S2	10.65	6.07	5.92	5.91	6.50	**5.81**
walk	S1	11.65	6.36	6.18	6.06	6.71	**5.93**
	S2	9.15	3.55	3.38	3.34	3.73	**3.15**
Average		12.23	7.01	6.90	6.85	7.91	**6.62**

Table 2. Evaluation of different algorithms in CMU motion capture database; the learning and inference time is also given in (*seconds*)

Error (cm)		KR	LR	MLR [4]	MoE [14]	GMR	Latent GMR
Subject 1	01-01	18.27	16.18	2.80	15.79	17.76	**14.49**
	01-05	22.27	23.01	22.05	21.77	20.36	**18.49**
	01-08	32.88	34.74	34.46	34.12	35.02	**32.76**
Subject 8	08-02	19.00	13.93	13.43	13.14	15.00	**11.78**
	08-03	17.59	19.95	19.34	19.17	19.90	**18.66**
	08-04	16.69	22.22	18.55	18.33	18.10	**15.45**
Subject 15	15-06	20.24	13.64	13.31	13.25	14.26	**13.14**
	15-11	15.90	14.26	13.81	13.59	14.88	**13.42**
	15-13	28.82	23.18	23.12	23.01	27.46	**22.95**
Subject 17	17-03	29.49	25.49	24.02	23.68	26.73	**23.23**
	17-05	21.43	13.93	13.70	13.42	15.96	**12.01**
	17-07	21.43	15.84	15.05	14.77	16.69	**14.55**
Average		21.99	19.68	18.83	18.61	20.13	**17.54**
Train time		0	0.06	75	79.18	17.82	19.38
Test time		10.28	0.02	0.16	0.17	14.89	1.14

Table 3. Detailed experiments on CMU motion capture database. We train on Subject 17, Sequences 01–05 and test on Subject 17, Sequences 07–10.

Error (cm)	KR	LR	MLR [4]		MoE [14]		GMR	
			Exp	B8	Exp	B8	Exp	B8
Orig. Space	37.98	25.69	26.73	23.40	25.51	23.26	29.75	20.74
PCA	43.21	35.18	25.04	22.33	24.81	22.22	25.14	16.37
LPP	43.57	22.75	22.70	21.45	22.61	21.29	23.15	**12.79**

both by running additional experiments on sub-set of data. In addition to prior experiments, we test PCA, as an alternative to LPP, for latent representation and a variety of regression models for inference within the latent spaces. We also show the performance of the expectation (Exp) as well as of multi-hypothesis mode prediction (B8) (assuming existence of oracle that chooses among the 8 mixture components). The results are illustrated in Table 3.

Based on Table 3 we make the following 4 observations: (1) inference in the latent space is nearly always better than in the original space; (2) LPP outperforms PCA in terms ability to preserve the manifold structure; (3) LR, MLR, MoE and GMR perform similarly on uni-modal prediction task; and (4) GMR outperforms all other methods with multiple predictions. We belive that (4) is due to the ability of GMR to generatively model the full density over the latent features and poses (as opposed to other more direct regression methods).

Finally, to compare to published methods we run on Agarwal and Triggs dataset [3], where we achieve error of 6.71 degrees, which is better the Nearest Neighbor regression and Linear Regression as reported in [17] and [18] respectively. However, we cannot match the performance of non-parametric shared LVMs, like Shared GPLVM and Shared KIE, that achieve errors of 6.50 and 5.95 degrees respectively. This is not surprising given that non-parametric models can represent more complex manifold structure; however, they do come at a cost of inference and learning which, unlike in our method, are a function of the training set size.

5 Conclusions and Future Work

In this paper, we present a parametric discriminative framework for 3D pose inference. Our model has a number of appealing properties, mainly: (1) it can deal with multi-modalities in the data, (2) model complex structure of the image feature and pose manifolds, (3) provides both forward and backwards mapping between the respective manifolds and original image feature or pose spaces, simplifying the inference and (4) alleviates the need for learning, a sometimes hard to obtain shared non-linear manifold structure. We show that our performance is comparative or superior to parametric and non-parametric models in the original high-dimensional space. In the future, we intend to look at learning the model in a unified manner through a single (as opposed to stage-wise) learning procedure.

References

1. Sidenbladh, H., Black, M., Fleet, D.: Stochastic tracking of 3D human figures using 2D image motion. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1843, pp. 702–718. Springer, Heidelberg (2000)
2. Sminchisescu, C., Triggs, B.: Covariance scaled sampling for monocular 3d body tracking. In: CVPR (2001)
3. Agarwal, A., Triggs, B.: 3d human pose from silhouettes by relevance vector regression. In: CVPR (2004)
4. Agarwal, A., Triggs, B.: Monocular human motion capture with a mixture of regressors. In: CVPR (2005)
5. Bissacco, A., Yang, M., Soatto, S.: Fast human pose estimation using appearance and motion via multi-dimensional boosting regression. In: CVPR (2007)
6. Bo, L., Sminchisescu, C.: Structured output-associative regression. In: CVPR (2009)

7. Elgammal, A.M., Lee, C.S.: Inferring 3d body pose from silhouettes using activity manifold learning. In: CVPR (2004)
8. Fathi, A., Mori, G.: Human pose estimation using motion exemplars. In: ICCV (2007)
9. Guo, F., Qian, G.: Learning and inference of 3d human poses from gaussian mixture modeled silhouettes. In: ICPR (2006)
10. Jaeggli, T., Koller-Meier, E., Van Gool, L.: Learning Generative Models for Multi-Activity Body Pose Estimation. International Journal of Computer Vision 83, 121–134 (2009)
11. Kanaujia, A., Sminchisescu, C., Metaxas, D.: Spectral latent variable models for perceptual inference. In: ICCV (2007)
12. Navaratnam, R., Fitzgibbon, A., Cipolla, R.: The Joint Manifold Model for Semi-supervised Multi-valued Regression. In: ICCV (2007)
13. Shakhnarovich, G., Viola, P., Darrell, T.: Fast pose estimation with parameter-sensitive hashing. In: ICCV (2003)
14. Sminchisescu, C., Kanaujia, A., Li, Z., Metaxas, D.: Discriminative density propagation for 3d human motion estimation. In: CVPR (2005)
15. Sminchisescu, C., Kanaujia, A., Metaxas, D.: Learning joint top-down and bottom-up processes for 3d visual inference. In: CVPR (2006)
16. Urtasun, R., Darrell, T.: Sparse probabilistic regression for activity-independent human pose inference. In: CVPR (2008)
17. Ek, C., Torr, P., Lawrence, N.: Gaussian process latent variable models for human pose estimation. In: Popescu-Belis, A., Renals, S., Bourlard, H. (eds.) MLMI 2007. LNCS, vol. 4892, pp. 132–143. Springer, Heidelberg (2008)
18. Sigal, L., Memisevic, R., Fleet, D.J.: Shared Kernel Information Embedding for Discriminative Inference. In: CVPR (2009)
19. He, X., Niyogi, P.: Locality preserving projections. In: NIPS (2003)
20. Nadaraya, E.: On estimation regression. Theory of Probability and its Applications 9, 141–142 (1964)
21. Sminchisescu, C., Kanaujia, A., Li, Z., Metaxas, D.: Learning to reconstruct 3d human motion from bayesian mixtures of experts: A probabilistic discriminative approach. Technical Report CSRG-502, University of Toronto (2004)
22. Thorndike, R.: Canonical correlation analysis. Applied Multivariate Statistics and Mathematical Modeling, 237–263 (2000)
23. Hotelling, H.: Relations between two sets of variates. Biometrika 28, 321–377 (1936)
24. Tian, T., Li, R., Sclaroff, S.: Articulated pose estimation in a learned smooth space of feasible solutions. In: Worshop on Learning in Computer Vision and Pattern Recognition, San Diego (2005)
25. E–frontier. Curious Labs Poser. Computer Software
26. CMU Motion Capture Database, http://mocap.cs.cmu.edu/
27. Sigal, L., Balan, A., Black, M.: Combined discriminative and generative articulated pose and non-rigid shape estimation. In: NIPS (2007)

Top-Down Cues for Event Recognition

Li Li[1,2], Chunfeng Yuan[1], Weiming Hu[1], and Bing Li[1]

[1] Institute of Automation, Chinese Academy of Sciences
[2] Radio, Film and Television Design and Research Institute

Abstract. How to fuse static and dynamic information is a key issue in event analysis. In this paper, we present a novel approach to combine appearance and motion information together through a top-down manner for event recognition in real videos. Unlike the conventional bottom-up way, attention can be focused volitionally on top-down signals derived from task demands. A video is represented by a collection of spatio-temporal features, called video words by quantizing the extracted spatio-temporal interest points (STIPs) from the video. We propose two approaches to build class specific visual or motion histograms for the corresponding features. One is using the probability of a class given a visual or motion word. High probability means more attention should be paid to this word. Moreover, in order to incorporate the negative information for each word, we propose to utilize the mutual information between each word and event label. High mutual information means high relevance between this word and the class label. Both methods not only can characterize two aspects of an event, but also can select the relevant words, which are all discriminative to the corresponding event. Experimental results on the TRECVID 2005 and the HOHA video corpus demonstrate that the mean average precision has been improved by using the proposed method.

1 Introduction

Event recognition is a key task in automatic video analysis, such as semantic summarization, annotation and retrieval. Since 2001, the National Institute of Standards and Technology (NIST) has started benchmarking content-based-video retrieval technologies, known as TRECVID [1], in which event detection is one of the evaluation tasks. NIST provides a benchmark of annotated video corpus for detecting a set of predefined concepts. Although a lot of efforts have been made for video based event recognition [2,3,4] and some preliminary results have been achieved during the past several years, the problem is still far away from being solved. This is mainly due to the within-event variations caused by many factors, such as unconstrained motions, cluttered backgrounds, occlusions, environmental illuminations and objects' geometric variances.

Recently, many researchers showed their interests in an approach that considers each video sequence as a collection of spatio-temporal interest points (STIPs). Laptev *et al.* [5] first incorporated the temporal constraint to a Harris interesting point detector to detect local 3D interesting points in the space-time dimension.

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 691–702, 2011.

Dollar *et al.* [6] improved the 3D Harris detector and applied Gabor filtering to the spatial and temporal domain to detect interest points. In this method, a video can be modeled by the Bag of Words(BoW) [7]model, which has ability to handle variability in viewpoints, illumination and scales. This influential model represents each video as a collection of independent codewords in a pre-defined codebook generated from the training data.

In a video clip, an event usually has two important attributes: *what* and *how*. The *what* attribute usually refers to the appearance information obtained from static images. SIFT features [8] have been proved to be good candidates for the representation of image static information. Similar to SIFT features, Dalal and Triggs [9] proposed Histogram of Oriented Gradient (HOG) descriptors to handle pedestrian detection in static images. Recently, Scovanner *et al.* [10] proposed 3D SIFT features by applying sub-histograms to encode local temporal and spatial information. On the other hand, the *how* attribute refers to an event's dynamic information usually the object's motion. Motion feature has always been considered as an important cue to characterize an event. For instance, in [11], the event is modeled by volumetric features derived from optical flow in a video sequence. Zhang [12] extracted motion templates (motion images and motion context) using very simple processing. Histogram of oriented optical flow (HOOF) [13]was used to recognize human actions by classifying HOOF time series. Although the above existing approaches partially solved the even recognition in different aspects, how to effectively combine both *what* and *how* attributes is still an open problem for event recognition. To address this issue, in [2], a set of methods with motion and bag-of-visual-words combination were proposed to exploit the relativeness of the motion information and the relatedness of the visual information. Dalal *et al.* [9] combined motion and HOG appearance to achieve more robust descriptor. Efros *et al.* [14] employed appearance and flow features in an exemplar based detector for long shots of sports players, though quantitative performance results were not given.

The conventional approach representing a video usually adopts a bottom-up paradigm. In this work, we choose a top-down human visual system [15] instead to combine visual and motion cues for event recognition. In this top-down human visual system, only a subset of interesting information will be focused while the rest will be demoted. In other words, not all spatio-temporal features make the same contribution for different types of events. Some features may be useless for a particular class of events. If more weights given to the features highly relevant to the event recognition, the performance could be improved. Therefore, we propose two approaches in order to weight features: (1) Firstly, the probability of each word w.r.t the event classes is computed. Then a class-specific histogram is constructed so that the STIPs with the higher probability of the corresponding words under the category should be emphasized more; (2)As an alternative to the probability, the mutual information (MI) of each word w.r.t the event classes is computed. Mutual information is a nonparametric, model-free method for scoring a set of features. It can be used to spot all features relevant to the classification, and to identify groups of features that allow building a

valid classification model. Recently, MI has been proved to be effective way for the computation of visual recognition tasks. Liu and Shah [16] utilized the Maximization of Mutual Information (MMI) to automatically discover the optimal number of video word clusters. Yuan *et al.* [17] represented the video sequence as a bag of spatio-temporal invariant points (STIPs), where the MI between each STIP and a specific class was evaluated. In this method, action categorization is based on the mutual information maximization. Due to the independence assumption of STIPs, this model ignored the dependency among features. Different from [17], in this paper, we calculate the MI between each word rather than each STIP and specific class. This MI considers not only the positive effect of each word, but also the negative tone. By incorporating the negative training information, our model gains better discriminative power.

The rest of the paper is organized as follows. Section 2 presents the overall architecture of the proposed method. Section 3 describes the proposed video representation method and the class specific histogram in details. Section 4 provides experiment results. Finally, conclusions are given in Section 5.

2 Overall Architecture

This paper focuses on developing effective techniques for the combination of visual and motion features. Although more complicated appearance features such as 3D SIFT could be used instead and may achieve even better results, we adopt a relatively simple features, HOG (Histogram of Oriented Gradient) as visual descriptors and HOF (Histogram of Optical Flow) as motion descriptors in order to validate the proposed combination methods.

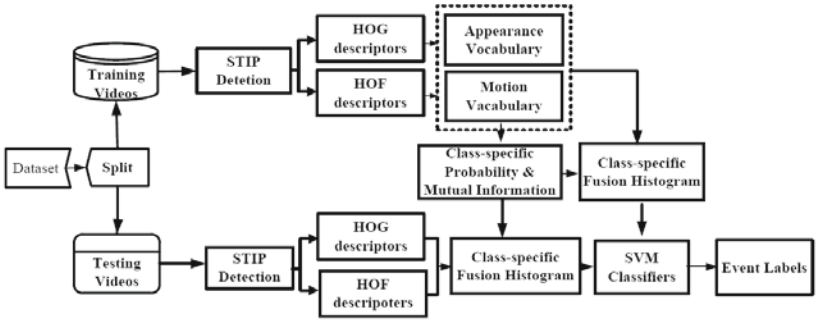


Fig. 1. The flowchart of the proposed approach for event recognition

Fig. 1 gives an overview of the framework. First of all, we employ Laptev *et al.* [5]'s method to detect spatio-temporal interest points (STIPs) in a video clip. After that, we utilize HOG as appearance descriptor and HOF as motion descriptor. Subsequently, the visual and motion codebook are generated respectively by grouping the detected STIP features using the k-means algorithm. The center of each resulting cluster is defined as a video word. Subsequently, in order to combine

visual and motion features, we build appearance based motion histogram and motion based appearance histogram in a top-down manner. For the learning process we use a method similar to [15]. We start with a set of training videos, in which all of the positive training sets have been manually marked. For example, HOG can be used as the descriptor cue and HOF can function as attention cue. Based on the probability and mutual information of the class for the given word, a class specific histogram is constructed. In this way, each video clip is eventually represented as an attention histogram in the framework of BoW. In the testing phase, the test video is also firstly represented as the attention histogram of BOW and then classified according to histogram matching between the test video and training videos. Finally, the test video is classified according to a SVM classifier with Histogram Intersection kernel.

In the next section, our descriptor and the attention histograms are described in details.

3 Top-Down Attention Histogram for Event Representation

3.1 STIPs: Video Representation

Bag-of-Words (BoW) [7,18] model has been proved to be a powerful tool for various image analysis tasks. The visual vocabulary provides a mid-level representation which helps to bridge the semantic gap between the low-level features and the high-level concepts. We represent a video as a bag of spatio-temporal features $\{d_i\}$. Once the visual and motion codebook are generated, we represent a video clip by $\nu = \{w^k\}$ and $k \in \{a, m\}$ for the two cues appearance and motion codebook respectively. $W \in \omega = \{w_1^k, w_2^k, ..., w_n^k\}$ represents a set of video words. We denote by $\mathrm{T}^{c+} = \{\nu_i\}$ the positive training samples of class c. Symmetrically, T^{c-} is the negative training dataset of class c, and $\mathrm{T} = \mathrm{T}^{c+} \cup \mathrm{T}^{c-}$. For a standard single-cue BoW, videos are represented by the statistical distribution of BoW:

$$n(w^k|\nu) = \sum_{j=1}^{\|\nu\|} \delta(w_{d_j}^k, w^k) \tag{1}$$

where $\|\nu\|$ denotes the total number of STIPs in video ν, $\delta(\cdot)$ is the indication function, $w_{d_j}^k$ is the word index of the corresponding STIP d_j. Conventionally fusion methods of the two cues are called early fusion and late fusion respectively, while early fusion involves creating one joint appearance-motion vocabulary, and late fusion concatenates both histogram representation of both appearance and motion, obtained independently.

3.2 Top-Down Attention Histogram

Inspired by the recent work [15], in which a top-down color attention mechanism combines the advantages of early and late feature fusion together, we resort to

the top-down human visual attention mechanism to recognize a specific event category $c \in \mathbf{C}$. Evidence from human vision indicates that high-level, class-based criteria play a crucial role in recognizing objects [19]. The computation of the video representation is done according to

$$n(w^a|\nu, \mathbf{C} = c) = \sum_{j=1}^{\|\nu\|} \pi(w^m_{d_j}, \mathbf{C} = c)\delta(w^a_{d_j}, w^a) \tag{2}$$

or

$$n(w^m|\nu, \mathbf{C} = c) = \sum_{j=1}^{\|\nu\|} \pi(w^a_{d_j}, \mathbf{C} = c)\delta(w^m_j, w^m) \tag{3}$$

where $\mathbf{C} = \{1, 2, .., C\}$ is the class label set. Eq. 2 and Eq. 3 indicate that the visual and motion information play predominant roles. $\pi(w^k_j, \mathbf{C} = c)$ is the attention information between feature w^k and class c. It will function as the weight of the other cue. For example, by Eq. 2, if motion is the predominant cue, we get an N-dimensional feature vector, where N is the number of visual words, and each element is a C-component of motion cue. Each component is the motion attention weight w.r.t class c. This attention based video representation indeed encodes both *what* and *how* aspects of an event. Each histogram is about the specific visual word which depicts *what* aspect, while the motion cue not only guides the impact of the visual word in capturing *how* aspect but also describes our prior knowledge about the categories we are looking for in the top-down manner. Similarly, by Eq. 2 the appearance information function as predominance cue is deployed to modulate the motion features. After concatenating the class-specific histogram, a video clip ν is eventually represented by a $N * C$-dimensional feature vector. In this paper, we propose two approaches to compute the attention information $\pi(w^k, \mathbf{C} = c)$: one is the probability of each word w^k w.r.t specific class c; the other is the mutual information between each word w^k and class c.

probabilistic vote. We resort to the probability for every word w.r.t the specific class to characterize the impact of the local features on the video representation.

$$\pi(w^k, \mathbf{C} = c) = P(\mathbf{C} = c|w^k) \tag{4}$$

Given a visual or motion word, the probability of a class c $P(\mathbf{C} = c|w^k)$ can be estimated by using Bayes formula,

$$P(\mathbf{C} = c|w^k) = \frac{P(w^k|\mathbf{C} = c)P(\mathbf{C} = c)}{P(w^k)} \tag{5}$$

where $P(w^k|\mathbf{C} = c)$ is the empirical distribution,

$$p(w^k|\mathbf{C} = c) = \frac{1}{\|\mathrm{T}^{c+}\|} \sum_{w^k_{d_j} \in \mathrm{T}^{c+}} \delta(w^k_{d_j}, w^k) \tag{6}$$

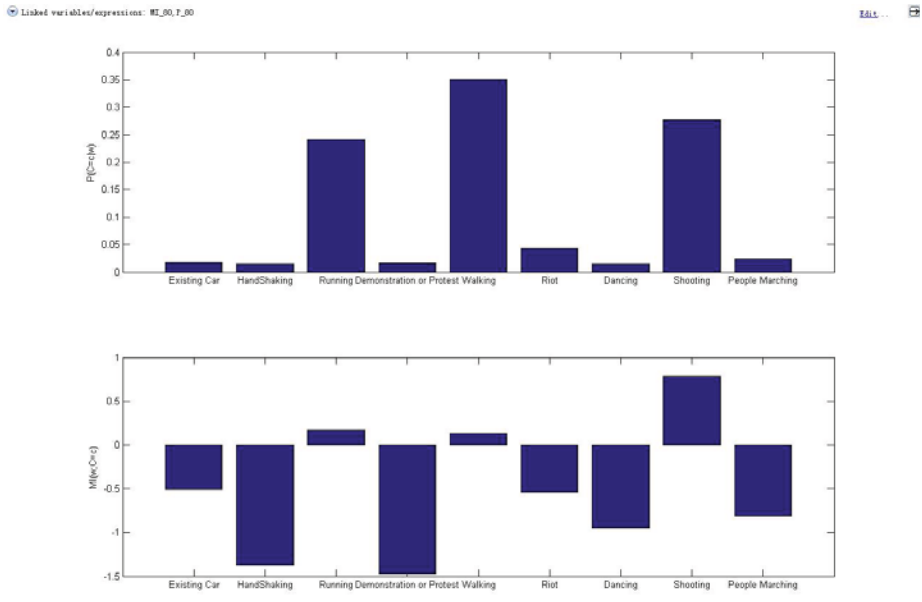


Fig. 2. Example class specific information given word w^k

can be obtained by summing over the indexes to the positive training videos in class c. $P(w^k)$ is the probability of word w^k in all training videos.

$$p(w^k) = \frac{1}{\|\mathrm{T}\|} \sum_{w^k_{d_j} \in \mathrm{T}} \delta(w^k_{d_j}, w^k) \tag{7}$$

High probability w.r.t specific class c means more attention could be paid to the corresponding given word. However, if the probability is almost equal for every class, this word will be regarded as irrelevant for the recognition task. By Eq. 3, 2 and Eq. 4, Motion Probability based Appearance Histogram (MPAH) and Appearance Probability based Motion Histogram (APMH) can be obtained, in which motion and appearance cues are predominant respectively.

Mutual Information vote. By Eq. 4 only the positive training information is take into consideration. In [17], better discriminative can be obtained by incorporating the negative information. Therefore, we resort to the mutual information to measure the importance of the features. However, we evaluate the mutual information between a word and a specific class $c \in \mathbf{C}$ rather than the mutual information between a STIP and a specific class, since the latter needs the independence assumption.

$$\pi(w^k, \mathbf{C} = c) = MI(w^k; c) \tag{8}$$

where $MI(w^k, c)$ is the mutual information between word w^k and class c can be obtained by

$$\begin{aligned} MI(w^k; c) &= \log \frac{P(w^k|\mathbf{C}=c)}{P(w^k)} \\ &= \log \frac{P(w^k|\mathbf{C}=c)}{P(w^k|\mathbf{C}=c)P(\mathbf{C}=c) + P(w^k|\mathbf{C}\neq c)P(\mathbf{C}\neq c)} \\ &= \log \frac{1}{P(\mathbf{C}=c) + \frac{P(w^k|\mathbf{C}\neq c)}{P(w^k|\mathbf{C}=c)}P(\mathbf{C}\neq c)} \end{aligned} \tag{9}$$

From Eq. 9, we can see that the likelihood ratio test $\frac{P(w^k|\mathbf{C}\neq c)}{P(w^k|\mathbf{C}=c)}$ determines whether w^k votes positively or negatively for class c. If $\frac{P(w^k|\mathbf{C}\neq c)}{P(w^k|\mathbf{C}=c)} > 1$, then $MI(w^k, c) < 0$, which means this video word w^k votes a negative score for the class c. On the contrary, when $\frac{P(w^k|\mathbf{C}\neq c)}{P(w^k|\mathbf{C}=c)} < 1$, then $MI(w^k, c) > 0$,w^k votes a positive score for the class c. The likelihood $\frac{P(w^k|\mathbf{C}\neq c)}{P(w^k|\mathbf{C}=c)}$ can be obtained by

$$\frac{P(w^k|\mathbf{C}\neq c)}{P(w^k|\mathbf{C}=c)} = \frac{\frac{1}{\|\mathrm{T}^{c-}\|}\sum_{w^k_{d_j}\in \mathrm{T}^{c-}} \delta(w^k_{d_j}, w^k)}{\frac{1}{\|\mathrm{T}^{c+}\|}\sum_{w^k_{d_j}\in \mathrm{T}^{c+}} \delta(w^k_{d_j}, w^k)} \tag{10}$$

From the representation of $MI(w^k, c)$ we can observe that both positive and negative training information vote a score for the class c. Similarly, $MI(w^k, c)$ encodes how much information from word w^k in class c. High mutual information between word w^k and class label c means that the word feature w^k is highly relevant. Fig. 2 shows an example of the class specific information. For a motion word w^k extracted from TRECIVD 2005 video dataset, both the probability of each class and the mutual information w.r.t each class c is computed. Note that, generally, high probability corresponding to high information. In this instance, given a word, the probability of class "Walking"is the maximum, while the mutual information $MI(w^m, \mathbf{C} = 5)$ is positive thus means this word is relate to the video event with class label "Walking". However, the mutual information is not necessarily the highest. In contrast, the words with mutual information near zero are statistically independent from the class label, where the ones with negative mutual information vote a negative score for the corresponding label.

4 Experiment

4.1 Data Sets

In order to demonstrate the performance of the proposed method, two different datasets :HOHA [5] and TRECVID 2005 [1], are used. Fig. 3 shows some sample pictures. HOHA contains 430 video clips, i.e., short sequences from 32 movies, of which 219 are used for training and 211 are used for testing. Each sample is

Some sample frames from the HOHA video dataset. From left to right: AnswerPhone, Kiss, SitUp, HandShake, SitDown, HugPerson, GetOutCar, StandUp

Some sample frames from the TRECVID 2005 video corpus. From left to right: Existing Car, Handshaking, Running, Demonstration or protest, Walking, Riot, Dancing, Shooting, People Marching.

Fig. 3. Example frames from HOHA and TRECVID dataset

annotated according to 8 classes: AnswerPhone, GetOutCar, HandShake, HugPerson, Kiss, SitDown, SitUp, StandUp. The ground truth of TRECVID 2005 dataset is based on LSCOM annotated events concepts [20]. After removing the events with a few positive samples, 9 of which are chosen as our evaluation set. Because the LSCOM annotation labels are deficient for our dynamic concepts, we re-annotated the event labels by watching all frames within the video shot. As a result, there are 1610 positive clips in total over the 9 events: Existing Car, Handshaking, Running, Demonstration Or Protest, Walking, Riot, Dancing, Shooting, People Marching, among which, half is used for classifier training and the remaining for testing. We evaluate the classification performance using the Average Precision (AP) measure, which is the standard evaluation metric employed in the TRECVID benchmark. Mean average precision (MAP) Average precision is proportional to the area under a recall-precision curve. To calculate AP for one concept, we first rank the test data according to the prediction of each sample. MAP is then calculated by averaging APs across all concepts.

4.2 Classifier

For classification, SVMs [21] are employed as "one-against-all" manner to estimate the likelihood of a given feature vector extracted from a video clip belonging to an event. In our experiments we use libSVM [21] with intersection kernel since it requires significantly less computational time while makes satisfactory results. For two BoW based histograms H_i and H_j extracted from video i and j, the intersection kernel is computed as:

$$K(H_i, H_j) = \frac{\sum_{n=1}^{N*C} \min(H_i(n), H_j(n))}{\min(\sum_{n=1}^{N*C} H_i(n), \sum_{n=1}^{n} H_j(n))} \tag{11}$$

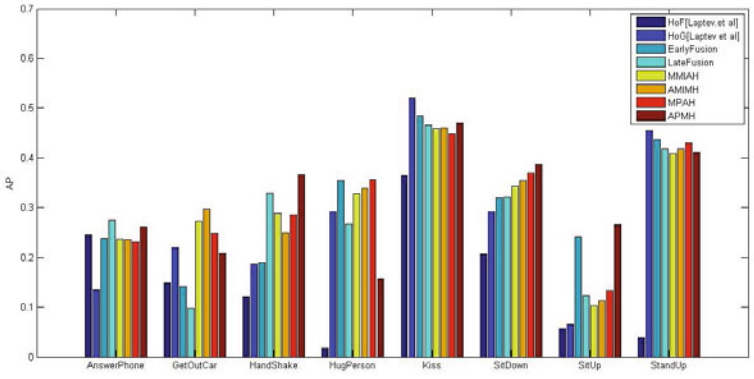


Fig. 4. Comparison of APs(%) between different features for recognizing action events in HOHA. HOG: Histogram of Oriented Gradient; HOF: Histogram of Flow; AMIMH: Appearance Mutual Information based Motion Histogram; MMIAH: Motion Mutual Information based Appearance Histogram; APMH: Appearance Probability based Motion Histogram; MPAH: Motion Probability based Appearance Histogram.

4.3 HOHA

In this section, we present the results using our proposed method on HOHA dataset. We build a vocabulary of 600 visual words, 600 motions words and 600 combination words by clustering the HOG and HOF and HOG+HOF descriptors respectively. Fig. 4 shows the performances for eight actions on HOHA. Unlike most existing approaches which need object tracking, detection or ground substraction, our method is data driven and therefore does not need any preprocessing step. Moreover, there is no parameters needed to be determined in our method. From Fig. 4, we can observe that appearance (HOG) and motion (HOF) itself could not be a good guide to the performance of combined detector. Specially, Appearance Probability based Motion Histogram (APMH) achieves high mean AP (MAP) of 31.5%. This shows 17% improvement compared to HOG [5] (MAP=27%). Appearance Probability based Motion Histogram (APMH) outperforms EarlyFusion and LateFusion, this validates the proposed approach that the Appearance Probability based Motion Histogram does guide the action recognition. Specially, for some action events such as GetOutCar, HandShake, HugPerson, SitDown and SitUp the attention based combination feature methods perform best, and for the event "HandShake" mutual information vote perform best. It also can be seen that the improvement of mutual information based features in this dataset is limited. The reason is that mutual information is inclined to select rare words by Eq. refequ:MI2. On the other hand, the *what* aspect usually refers to a person in HOHA dataset, which means that the appearance features may not play a predominant role such that the appearance attention based motion histograms do not perform as good as motion attention based visual histograms.

Table 1. Comparison of Average Precision (%) using different features on TRECVID dataset

Event Name	HOG	HOF	HOG+HOF	LateFusion	MPAH	APMH	MMIAH	AMIMH
Existing-car	15.6	21.8	23.2	24.6	20.8	25.0	**30.3**	27.8
Handshaking	46.1	46.3	50.8	51.5	47.7	46.4	49.8	**54.2**
Running	76.0	76.7	78.3	76.3	76.6	77.6	76.4	**78.7**
Demonstration-Protest	26.4	25.0	16.3	25.2	26.2	26.8	**28.1**	27.5
Walking	72.0	71.2	72.3	**72.9**	72.1	72.0	70.9	72.2
Riot	28.9	27.2	28.0	28.2	30.1	28.8	30.6	**30.6**
Dancing	20.7	21.4	14.4	22.8	22.1	23.4	24.9	**26.3**
Shooting	60.1	65.6	65.3	62.2	61.8	65.1	**68.3**	65.7
People-Marching	21.9	20.6	22.6	21.0	21.1	21.9	**26.4**	24.1
Mean Average Precision	40.9	41.8	41.2	42.9	42.1	43.0	45.1	**45.2**

4.4 TRECVID

We also quantitatively compare the event recognition accuracy by using the proposed algorithm with different features. Table. 1 presents the performances of nine events on TRECVID video corpus. As shown, we have a set of interesting observations:

1. Among these features, the best performance gain is obtained by Appearance Mutual Information based Attention (AMIMH) with the highest MAP of 45.2%, this shows 10.5% improvement compared with HOG (MAP=40.9%). The reason is that HOG only captures *what* aspect of an event, but ignores *how* aspect. Similarly, compared with HOF, an improvement of 8.13% is achieved in that HOF only captures *how* aspect, but ignores *what* ones. Both HoG+HoF and Late Fusion outperform HOG and HOF, which shows the necessity of the combination of these two cues. For the latter four features such as MPAH and MMIAH, appearance words capture *what* aspect, while their corresponding motion class specific histograms not only describe *how* aspect but also provide more relevant motion features for specific class.

2. In general, the attention based features outperform conventional multiple feature combination methods such as early fusion or late fusion strategies. Unlike HOHA dataset, mutual information based attention approaches perform better than probability based attention in TRECVID dataset, which supports our argument in Section 3 that MI provides more discriminative features for the classification tasks by incorporating the negative votes of each word. The object are different from HOHA dataset, whereas former including person, car or other scenes.

3. A slight disappointing results of Motion Probability based Appearance Histogram compared with LateFusion method may be caused by the confusion motion probability of the given word. Such as for the event Existing car, the appearance of different cars are various, LateFusion has the property of "vocabulary compactness" [15], whereas the Motion Probability based Appearance Histogram lack it.

5 Conclusion

In this paper, we propose a novel approach by combining motion and visual features together for event recognition in Bag-of Words (BOW) framework. Given a visual or motion word, both the probability and mutual information of each class are used to guide the recognition in a top-down way. The results from TRECVID and HOHA dataset suggest that for most event categories attention based histograms not only capture two event aspects but also provide more discriminative features. Specially, no parameter needed to be determined within our approach.

References

1. http://www-nlpir.nist.gov/projects/trecvid
2. Wang, F., Jiang, Y.G., Ngo, C.W.: Video event detection using motion relativity and visual relatedness. In: Proceeding of the 16th ACM International Conference on Multimedia (2008)
3. Xu, D., Chang, S.F.: Video event recognition using kernel methods with multilevel temporal alignment. IEEE Trans. Pattern Analysis and Machine Intelligence 30 (2008)
4. Zhou, X., Zhuang, X., Yan, S., Chang, S.F., Hasegawa-Johnson, M., Huang, T.S.: Sift-bag kernel for video event analysis. In: Proceeding of the 16th ACM International Conference on Multimedia (2008)
5. Laptev, I., Marszałek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: IEEE Conference on Computer Vision and Pattern Recognition (2008)
6. Dollr, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking ans Surveillance, pp. 65–72 (2005)
7. Jiang, Y.G., Ngo, C.W., Yang, J.: Towards optimal bag-of-features for object categorization and semantic video retrieval. In: Proceedings of ACM International Conference on Image and Video Retrieval, vol. 46 (2007)
8. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60, 91–110 (2004)
9. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 886–893 (2005)
10. Paul Scovanner, S.A., Shah, M.: A 3-dimensional sift descriptor and its application to action recognition. In: Proceeding of the 15th ACM International Conference on Multimedia (2007)
11. Ke, Y., Sukthankar, R., Hebert, M.: Efficient visual event detection using volumetric features. In: International Conference on Computer Vision, pp. 166–173 (2005)
12. Zhang, Z., Hu, Y., Chan, S., Chia, L.-T.: Motion Context: A New Representation for Human Action Recognition. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 817–829. Springer, Heidelberg (2008)
13. Chaudhry, R., Ravichandran, A., Hager, G., Vidal, R.: IEEE Conference on Computer Vision and Pattern Recognition (2008)

14. Efros, A.A., Berg, A.C., Mori, G., Malik, J.: Recognizing action at a distance. In: IEEE International Conference on Computer Vision, pp. 726–733
15. Khan, F.S., van de Weijer, J., Vanrell, M.: Top-down color attention for object recognition. In: International Conference on Computer Vision (2009)
16. Liu, J., Shah, M.: Learning human actions via information maximization. In: IEEE Conference on Computer Vision and Pattern Recogntion, pp. 1996–2003 (2008)
17. Yuan, J., Liu, Z., Wu, Y.: Discriminative subvolume search for efficient action detection. In: IEEE Conference on Computer Vision and Pattern Recogntion, pp. 2442–2449 (2009)
18. Niebles, J.C., Wang, H., Fei-fei, L.: Unsupervised learning of human action categories using spatial temporal words. In: Proceedings of British Machine Vision Conference, pp. 299–318 (2006)
19. Chen, X., Zelinsky, G.J.: Real-world visual search is dominated by top-down guidance. Vision Research 46, 4118–4133 (2006)
20. DTO Challenge Workshop on Large Scale Concept Ontology for Multimedia. Revison of lscom event/activity annotations Columbia University ADVENT Technical Report (2006)
21. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), http://www.csie.ntu.edu.tw/~cjlin/libsvm

Robust Photometric Stereo via Low-Rank Matrix Completion and Recovery

Lun Wu[1], Arvind Ganesh[2], Boxin Shi[4], Yasuyuki Matsushita[3], Yongtian Wang[1], and Yi Ma[2,3]

[1] School of Optics and Electronics, Beijing Institute of Technology, Beijing
[2] Coordinated Science Lab, University of Illinois at Urbana-Champaign
[3] Visual Computing Group, Microsoft Research Asia, Beijing
[4] Key Laboratory of Machine Perception, Peking University, Beijing
lun.wu@hotmail.com, abalasu2@illinois.edu, shiboxin@cis.pku.edu.cn, yasumat@microsoft.com, wyt@bit.edu.cn, mayi@microsoft.com

Abstract. We present a new approach to robustly solve photometric stereo problems. We cast the problem of recovering surface normals from multiple lighting conditions as a problem of recovering a low-rank matrix with both missing entries and corrupted entries, which model all types of non-Lambertian effects such as shadows and specularities. Unlike previous approaches that use Least-Squares or heuristic robust techniques, our method uses advanced convex optimization techniques that are guaranteed to find the correct low-rank matrix by simultaneously fixing its missing and erroneous entries. Extensive experimental results demonstrate that our method achieves unprecedentedly accurate estimates of surface normals in the presence of significant amount of shadows and specularities. The new technique can be used to improve virtually any photometric stereo method including uncalibrated photometric stereo.

1 Introduction

Photometric stereo [1,2] estimates surface orientations from photographs taken from a fixed viewpoint under different lighting conditions. Since photometric stereo can produce a dense normal field at the level of detail that cannot be achieved by any other triangulation-based approaches, it has generated a lot of interest for accurate shape reconstruction.

It is well understood that when a Lambertian surface is illuminated by at least three known lighting directions, the surface orientation at each visible point can be uniquely determined from its intensities. From different perspectives, it has long been shown that if there are no shadows, the appearance of a convex Lambertian scene illuminated from different lighting directions span a three-dimensional subspace [3] or an illumination cone [4]. Basri and Jacobs [5] and Georghiades *et al* [6] have further shown that the images of a convex-shaped object with cast shadows can also be well-approximated by a low-dimensional linear subspace. The aforementioned works indicate that there exists a degenerate structure in the appearance of Lambertian surfaces under variation in illumination. This is the key property that all photometric stereo methods harness to determine the surface normals.

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 703–717, 2011.

Previously, photometric stereo algorithms for Lambertian surfaces generally find surface normals as the *Least Squares* solution to a set of linear equations that relate the observations and known lighting directions, or equivalently, try to identify the low-dimensional subspace using conventional Principal Component Analysis (PCA) [7]. Such a solution is known to be optimal if the measurements are corrupted by only *i.i.d.* Gaussian noise of small magnitude. Unfortunately, in reality, photometric measurements rarely obey such a simplistic noisy linear model: the intensity values at some pixels can be severely affected by specular reflections (deviation from the basic Lambertian assumption), sensor saturations, or shadowing effects. As a result, the Least Squares solution normally ends up with incorrect estimates of surface orientations in practice.

To overcome this problem, researchers have explored various approaches to eliminate such deviations by treating the corrupted measurements as outliers, *e.g.*, using a RANSAC scheme [8,9], or a median-based approach [10]. To identify the different types of corruptions in images more carefully, Mukaiegawa *et al* [11] have proposed a method for classifying diffuse, specular, attached, and cast shadow pixels based on RANSAC and outlier elimination.

Contributions: In this paper, we propose a simple but principled solution to photometric stereo that can deal with any kind of deviation from the basic Lambertian assumption in a unified framework. We cast the photometric stereo problem as a problem of recovering and completing a low-rank matrix subject to sparse, gross errors like corrupted and missing pixels. Unlike previous heuristic methods, under fairly broad conditions, the new method is guaranteed to correctly recover the low-rank Lambertian diffuse component from the highly corrupted and incomplete observations. Based on advanced convex optimization tools for nuclear norm and ℓ_1-norm minimization, the new method can efficiently obtain highly accurate estimates of surface orientations. Our method can be used to improve virtually any existing photometric stereo method, including uncalibrated photometric stereo [12], where traditionally, corruption in the data (*e.g.*, by specularity) is either neglected or ineffectively dealt with conventional heuristic robust estimation methods.

In contrast to previous robust approaches, our method is computationally more efficient and provides theoretical guarantees for robustness to large errors. More importantly, our method is able to use all the available information simultaneously for obtaining the optimal result, instead of discarding informative measurements, *e.g.*, by either selecting the best set of illumination directions [9] or using the median estimator [10].

2 Photometric Stereo as Low-Rank Matrix Recovery with Sparse Errors

In this section, we formulate the problem of estimating the normal map as a rank minimization problem. We first review the basic Lambertian image formation model, and then discuss how to model large deviations like shadows and specularities. In the following discussion, we make a few assumptions:

– The relative position of the camera and object is fixed across all images.
– The object is illuminated by a point light source at infinity.
– The sensor response is linear.

Lambertian Image Formation Model. The appearance I of a Lambertian scene observed under a lighting direction $\mathbf{l} \in \mathbb{R}^3$ is described as the inner product:

$$I = \rho\, \mathbf{n} \cdot \mathbf{l} \tag{1}$$

where ρ is the diffuse albedo, and $\mathbf{n} \in \mathbb{R}^3$ is the surface normal. Suppose that we are given n images $I_1, \ldots, I_n$ of a scene under different lighting conditions. Let the region of interest be composed of m pixels in each image.[1] We order the pixel locations with a single index k, and let $I_j(k)$ denote the observed intensity at pixel location k in image I_j. With this notation, we have the following relation about the observation $I_j(k)$:

$$I_j(k) = \rho_k\, \mathbf{n}_k \cdot \mathbf{l}_j \tag{2}$$

where ρ_k is the albedo of the scene at pixel location k, $\mathbf{n}_k \in \mathbb{R}^3$ is the (unit) surface normal of the scene at pixel location k, and $\mathbf{l}_j \in \mathbb{R}^3$ represents the normalized lighting direction vector corresponding to image I_j.[2] We assume that the light intensity is constant across images to simplify the discussion, although the proposed method is not limited to such a condition.

Low-rank Matrix Structure. Consider the matrix $D \in \mathbb{R}^{m \times n}$ constructed by stacking all the vectorized images $\mathrm{vec}(I)$ as

$$D = [\mathrm{vec}(I_1) \mid \cdots \mid \mathrm{vec}(I_n)] \tag{3}$$

where $\mathrm{vec}(I_j) = [I_j(1), \ldots, I_j(m)]^T$ for $j = 1, \ldots, n$. It follows from Eq. 2 that D can be factorized as follows:

$$D = NL \tag{4}$$

where $N \doteq [\rho_1 \mathbf{n}_1 \mid \cdots \mid \rho_m \mathbf{n}_m]^T \in \mathbb{R}^{m \times 3}$, and $L \doteq [\mathbf{l}_1 \mid \cdots \mid \mathbf{l}_n] \in \mathbb{R}^{3 \times n}$. Suppose that the number of images $n \geq 3$. Clearly, irrespective of the number of pixels m and the number of images n, the rank of the matrix D is at most 3.

Modeling Corruptions as Sparse Errors. The low-rank structure of the observation matrix D described above is seldom observed with real images. This is due to the presence of shadows and specularities in real images.

– **Shadows** arise in real images in two possible ways. Some pixels are not visible in the image because they face away from the light source. Such dark pixels are referred to as *attached shadows* [13]. In deriving Eq. 4 from Eq. 2, we have implicitly assumed that all pixels of the object are illuminated by the light source in each image. However, if the pixel faces away from the light source, then the relation no longer holds. Mathematically, this implies that Eq. 2 must be rewritten as follows:

[1] Typically, m is much larger than the number of images n.
[2] The convention here is that the lighting direction vectors point from the surface of the object to the light source.

$$I_j(k) = \max\{\rho_k \, \mathbf{n}_k \cdot \mathbf{l}_j, 0\} \tag{5}$$

Shadows can also occur in images when the shape of the object's surface is not convex: parts of the surface can be occluded from the light source by other parts. Even though the normal vectors at such occluded pixels may form a sharp angle with the lighting direction, these pixels appear entirely dark. We refer to such dark pixels as *cast shadows*. Irrespective of the type, all shadows occur in images as dark pixels with very small, if not zero, intensity values.

– **Specularities.** Specular reflection arises when the object of interest is not perfectly diffusive, *i.e.*, when the surface luminance is not purely isotropic. Thus, the intensity of reflected light depends on the viewing angle, and light is reflected in a mirror-like fashion accompanied by a specular lobe when viewed from certain angles. This gives rise to some bright spots or shiny patches on the surface of the object that significantly deviate from the Lambertian assumption.

Suppose we represent all these deviations from the ideal low-rank diffusive model Eq. 4 by an error matrix $E \in \mathbb{R}^{m \times n}$. Thus, instead of Eq. 4, the image measurements should be modeled as

$$D = NL + E \tag{6}$$

where the matrix E accounts for corruption by shadows or specularities. Now suppose that only a small fraction of the pixels in each image exhibit strong specular reflectance and that a large majority of the pixels are illuminated by the light source. Then, most pixels in the input images obey the low-rank diffusive model given by Eq. 4, and hence, most entries in the error matrix E will be zero, *i.e.*, E is a sparse matrix. If the matrix L of lighting directions is known, then we can compute the surface normals, provided that we can decompose D as the sum of a low-rank matrix and a sparse error matrix. Thus, the problem can be stated more formally as follows:

Let $I_1, \ldots, I_n$ be n images of an object under different illumination conditions. If $D \in \mathbb{R}^{m \times n}$ is defined as given in Eq. 3, then find a sparse matrix E such that the matrix $A \doteq D - E$ has the lowest possible rank.

Using a Lagrangian formulation, we can write the above problem as the following optimization problem:

$$\min_{A,E} \operatorname{rank}(A) + \gamma \, \|E\|_0 \quad \text{s.t.} \quad D = A + E \tag{7}$$

where $\|\cdot\|_0$ denotes the ℓ_0-norm (number of non-zero entries in the matrix), and $\gamma > 0$ is a parameter that trades off the rank of the solution A versus the sparsity of the error E. Let $(\hat{A}, \hat{E})$ be the optimal solution to Eq. 7. If the lighting directions L are given, we can easily recover the matrix N of surface normals from $\hat{A}$ as:

$$N = \hat{A} L^{\dagger} \tag{8}$$

where $L^{\dagger}$ denotes the Moore-Penrose pseudo-inverse of L. The surface normals $\mathbf{n}_1, \ldots, \mathbf{n}_m$ can be estimated by normalizing each row of N to have unit norm.

While Eq. 7 follows from our formulation, it is not tractable since both rank and ℓ_0-norm are non-convex and discontinuous functions. Solving this optimization problem efficiently will be the topic of discussion in the next section.

3 Efficient Solution via Convex Programming

As discussed above, the optimization problem given in Eq. 7 is extremely difficult (NP-hard in general) to solve. In this section, we propose to solve it efficiently based on recent advances in algorithms for matrix rank minimization [14,15,16].

3.1 Convex Relaxation and Modification

Recently, Wright *et al* [14] and Chandrasekaran *et al* [15] have proposed that the problem in Eq. 7 can be solved by replacing the cost function with its convex surrogate, provided that the rank of the matrix A is not too high and the number of non-zero entries in the matrix E is not too large. This convex relaxation, dubbed *Principal Component Pursuit* (PCP) in [14], replaces rank$(\cdot)$ with the *nuclear norm* (sum of the singular values of the matrix) and the ℓ_0-norm with the matrix ℓ_1-norm (sum of the absolute values of all entries of the matrix). Under quite general conditions, it has been proved in [14,15] that the following optimization problem has the same optimal solution as Eq. 7:

$$\min_{A,E} \|A\|_* + \lambda \|E\|_1 \quad \text{s.t.} \quad D = A + E \tag{9}$$

where $\|\cdot\|_*$ and $\|\cdot\|_1$ represent the nuclear norm and ℓ_1-norm, respectively, and $\lambda > 0$ is a weighting parameter. Theoretical considerations in [14] suggest that λ must be of the form $C/\sqrt{\max\{m,n\}}$, where C is a constant, typically set to unity. It is interesting to note that the equivalence between Eq. 7 and Eq. 9 is not affected by the magnitude of the singular values of the solution A or by the magnitude of the non-zero entries of the error matrix E.

In the framework of PCP, the locations of the non-zero entries in the sparse matrix E are assumed to be unknown a priori. But if the locations of some of the corrupted entries are known, then we can incorporate that information into the recovery procedure and hence, make the problem somewhat easier to solve. This is similar in spirit to the matrix completion problem [17,18,19]. Notice that although both shadows and specularities corrupt the low-rank matrix, they have different characteristics. While the locations of the specular pixels are hard to detect, especially that of pixels in specular lobes, it is relatively easy to detect the location of shadows in an image (*e.g.*, by a simple thresholding of the pixel values). Thus, we have more information about the shadows than specularities, and such information can greatly help finding the correct solution. So mathematically, we have a problem of recovering a low-rank matrix with both missing entries (the shadows) and unknown corrupted entries (the specularities).

We denote by Ω the locations of missing entries in the observed matrix D, defined in Eq. 3, that correspond to shadows in the input images. By a slight abuse of notation, we also denote by Ω the linear subspace of $m \times n$ matrices with

support in Ω. Let π_Ω represent the orthogonal projection operator corresponding to the subspace Ω. Thus, we modify the PCP problem in Eq. 9 to the following one which does both matrix completion and error correction:

$$\min_{A,E} \|A\|_* + \lambda \|E\|_1 \quad \text{s.t.} \quad \pi_{\Omega^c}(D) = \pi_{\Omega^c}(A+E) \tag{10}$$

where Ω^c denotes the linear subspace complementary to Ω, and π_{Ω^c} is the associated projection operator. The above problem is almost identical to the PCP problem (Eq. 9), except that the linear equality constraint is now applied only on the set Ω^c of pixels that are not affected by the detected shadows.

3.2 Fast Algorithm Using Augmented Lagrange Multiplier

The optimization problem in Eq. 10 can be re-cast as a semidefinite program and solved using interior-point methods. Although interior-point methods have excellent convergence properties, they are not very scalable for large problems. Fortunately, there has been a flurry of work recently on developing scalable algorithms for high-dimensional nuclear-norm minimization [16, 20, 21]. In this section, we show how one such algorithm, the Augmented Lagrange Multiplier (ALM) method [16, 22], can be adapted to efficiently solve Eq. 10.

The basic idea of the ALM method is to minimize the augmented Lagrangian function instead of directly solving the original constrained optimization problem. For our problem Eq. 10, the augmented Lagrangian is given by

$$\mathcal{L}_\mu(A,E,Y) = \|A\|_* + \lambda\|E\|_1 + \langle Y, \pi_{\Omega^c}(D-A-E)\rangle + \frac{\mu}{2}\|\pi_{\Omega^c}(D-A-E)\|_F^2 \tag{11}$$

where $Y \in \mathbb{R}^{m\times n}$ is a Lagrange multiplier matrix, μ is a positive constant, $\langle\cdot,\cdot\rangle$ denotes the matrix inner product,[3] and $\|\cdot\|_F$ denotes the Frobenius norm. For appropriate choice of the Lagrange multiplier matrix Y and sufficiently large constant μ, it can be shown that the augmented Lagrangian function has the same minimizer as the original constrained optimization problem [22]. The ALM algorithm iteratively estimates both the Lagrange multiplier and the optimal solution. The basic ALM iteration is given by

$$\begin{cases} (A_{k+1}, E_{k+1}) = \arg\min_{A,E} \mathcal{L}_{\mu_k}(A,E,Y_k) \\ Y_{k+1} = Y_k + \mu_k\, \pi_{\Omega^c}(D - A_{k+1} - E_{k+1}) \\ \mu_{k+1} = \rho \cdot \mu_k \end{cases} \tag{12}$$

where $\{\mu_k\}$ is a monotonically increasing positive sequence ($\rho > 1$).

We now focus our attention on solving the non-trivial first step of the above iteration. Since it is difficult to minimize $\mathcal{L}_{\mu_k}(\cdot)$ with respect to both A and E simultaneously, we adopt an alternating minimization strategy as follows:

$$\begin{cases} E_{j+1} = \arg\min_E \lambda\|E\|_1 - \langle Y_k, \pi_{\Omega^c}(E)\rangle + \frac{\mu_k}{2}\|\pi_{\Omega^c}(D - A_j - E)\|_F^2 \\ A_{j+1} = \arg\min_A \|A\|_* - \langle Y_k, \pi_{\Omega^c}(A)\rangle + \frac{\mu_k}{2}\|\pi_{\Omega^c}(D - A - E_{j+1})\|_F^2 \end{cases} \tag{13}$$

[3] $\langle X, Y\rangle \doteq \text{trace}(X^T Y)$.

Algorithm 1. (Matrix Completion and Recovery via ALM)

INPUT: $D \in \mathbb{R}^{m\times n}$, $\Omega \subset \{1,\ldots,m\}\times\{1,\ldots,n\}$, $\lambda > 0$.
Initialize $A_1 \leftarrow 0$, $E_1 \leftarrow 0$, $Y_1 \leftarrow 0$.
while not converged $(k = 1, 2, \ldots)$ **do**
  $A_{k,1} = A_k$, $E_{k,1} = E_k$;
  while not converged $(j = 1, 2, \ldots)$ **do**
    $E_{k,j+1} = \text{shrink}\left(\pi_{\Omega^c}(D) + \frac{1}{\mu_k}Y_k - \pi_{\Omega^c}(A_{k,j}), \frac{\lambda}{\mu_k}\right)$;
    $t_1 = 1$; $Z_1 = A_{k,j}$; $A_{k,j,1} = A_{k,j}$;
    while not converged $(i = 1, 2, \ldots)$ **do**
      $(U_i, \Sigma_i, V_i) = \text{svd}\left(\frac{1}{\mu_k}Y_k + \pi_{\Omega^c}(D) - E_{k,j+1} + \pi_{\Omega}(Z_i)\right)$;
      $A_{k,j,i+1} = U_i\,\text{shrink}\left(\Sigma_i, \frac{1}{\mu_k}\right)\,V_i^T$, $t_{i+1} = 0.5\left(1+\sqrt{1+4t_i^2}\right)$;
      $Z_{i+1} = A_{k,j,i+1} + \frac{t_i-1}{t_{i+1}}(A_{k,j,i+1} - A_{k,j,i})$, $A_{k,j+1} = A_{k,j,i+1}$;
    end while
    $A_{k+1} = A_{k,j+1}$; $E_{k+1} = E_{k,j+1}$;
  end while
  $Y_{k+1} = Y_k + \mu_k\,\pi_{\Omega^c}D - A_{k+1} - E_{k+1})$, $\mu_{k+1} = \rho\cdot\mu_k$;
end while
OUTPUT: $(\hat{A}, \hat{E}) = (A_k, E_k)$.

Without loss of generality, we assume that the Y_k's and the E_k's (and hence, Y and E, respectively) have their support in Ω^c. Then, the above minimization problems in Eq. 13 can be solved as described below.
We first define the *shrinkage* (or soft-thresholding) operator for scalars as follows:

$$\text{shrink}(x, \alpha) = \text{sign}(x)\cdot\max\{|x| - \alpha, 0\} \tag{14}$$

where $\alpha \geq 0$. When applied to vectors or matrices, the shrinkage operator acts element-wise. Then, the first step in Eq. 13 has a closed-form solution given by

$$E_{j+1} = \text{shrink}\left(\pi_{\Omega^c}(D) + \frac{1}{\mu_k}Y_k - \pi_{\Omega^c}(A_j), \frac{\lambda}{\mu_k}\right) \tag{15}$$

Since it is not possible to express the solution to the second step in Eq. 13 in closed-form, we adopt an iterative strategy based on the Accelerated Proximal Gradient (APG) algorithm [23,21,20] to solve it. The iterative procedure is given as:

$$\begin{cases} (U_i, \Sigma_i, V_i) = \text{svd}\left(\frac{1}{\mu_k}Y_k + \pi_{\Omega^c}(D) - E_{j+1} + \pi_{\Omega}(Z_i)\right) \\ A_{i+1} = U_i\,\text{shrink}\left(\Sigma_i, \frac{1}{\mu_k}\right)\,V_i^T \\ Z_{i+1} = A_{i+1} + \frac{t_i-1}{t_{i+1}}(A_{i+1} - A_i) \end{cases} \tag{16}$$

where $\text{svd}(\cdot)$ denotes the singular value decomposition operator, and $\{t_i\}$ is a positive sequence satisfying $t_1 = 1$ and $t_{i+1} = 0.5\left(1+\sqrt{1+4t_i^2}\right)$. The entire algorithm to solve Eq. 10 has been summarized as Algorithm 1.

4 Experiments

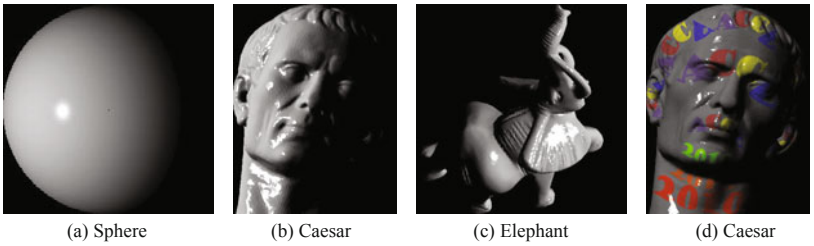
In this section, we verify the effectiveness of the proposed method using both synthetic and real-world images. We compare our results with a simple Least Squares (LS) approach, which assumes the ideal diffusive model given by Eq. 4. However, we do not use those pixels that were classified as shadows (the set Ω). Thus, the LS method can be summarized by the following optimization problem:

$$\min_{N} \|\pi_{\Omega^c}(D - N\,L)\|_F \tag{17}$$

We first test our algorithm using synthetic images whose ground-truth normal maps are known [24]. In these experiments, we quantitatively verify the correctness of our algorithm by computing the angular errors between the estimated normal map and the ground-truth. We then test our algorithm on more challenging real images. Throughout this section, we denote by m the number of pixels in the region of interest in each image, and by n the number of input images (typically, $m \gg n$).

4.1 Quantitative Evaluation with Synthetic Images

In this section, we use synthetic images of three different objects (see Fig. 1(a)-(c)) under different scenarios to evaluate the performance of our algorithm. Since these images are free of any noise, we use a pixel threshold value of zero to detect shadows in the images. Unless otherwise stated, we set $\lambda = 1/\sqrt{m}$ in Eq. 10.

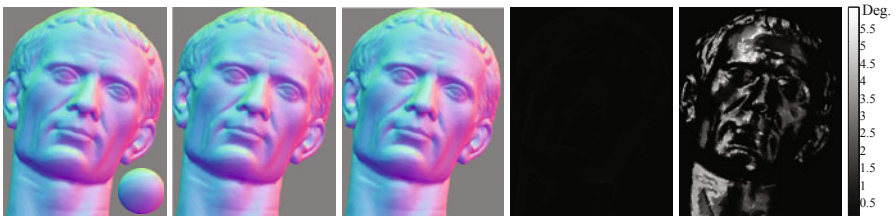


(a) Sphere (b) Caesar (c) Elephant (d) Caesar (with texture)

Fig. 1. Synthetic images used for experiments

a. Specular scene. In this experiment, we generate images of an object under 40 different lighting conditions, where the lighting directions are chosen at random from a hemisphere with the object placed at the center. The images are generated with some specular reflection. For all our experiments, we use the Cook-Torrance reflectance model [25] to generate images with specularities. Thus, there are two sources of corruption in the images – attached shadows and specularities.

A quantitative evaluation of our method and the Least Squares approach is presented in Table 1. The estimated normal maps are shown in Fig. 2(b),(c). We use the RGB channel to encode the 3 spatial components (XYZ) of the normal



(a) Ground truth (b) Our method (c) Least Squares (d) Error map (our method) (e) Error map (LS)

Fig. 2. Specular scene. 40 different images of Caesar were generated using the Cook-Torrance model for specularities. (a) Ground truth normal map with reference sphere. (b) and (c) show the surface normals recovered by our method and LS, respectively. (d) and (e) show the pixel-wise angular error w.r.t. the ground truth.

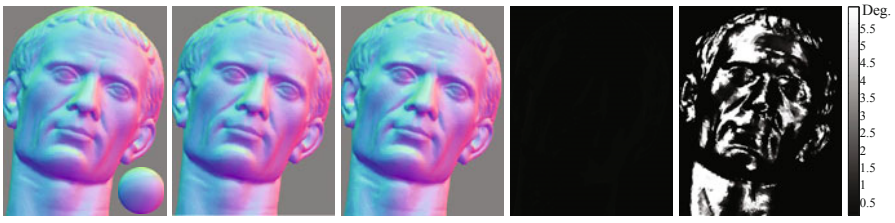
Table 1. Specular scene. Statistics of angle error in the normals for different objects. In each case, 40 images were used. In the rightmost column, we indicate the average percentage of pixels corrupted by attached shadows and specularities in each image.

Object	Mean error (in degrees)		Max. error (in degrees)		Avg. % of corrupted pixels	
	LS	Our method	LS	Our method	Shadow	Specularity
Sphere	0.99	$\mathbf{5.1 \times 10^{-3}}$	8.1	**0.20**	18.4	16.1
Caesar	0.96	$\mathbf{1.4 \times 10^{-2}}$	8.0	**0.22**	20.7	13.6
Elephant	0.96	$\mathbf{8.7 \times 10^{-3}}$	8.0	**0.29**	18.1	16.5

map for display purposes. The error is measured in terms of the angular difference between the ground truth normal and the estimated normal at each pixel location. The pixel-wise error maps are shown in Fig. 2(d),(e). From the mean and the maximum angular error (in degrees) in Table 1, we see that our method is much more accurate than the LS approach. This is because specularities introduce large magnitude errors to a small fraction of pixels in each image whose locations are unknown. The LS algorithm is not robust to such corruptions while our method can correct these errors and recover the underlying rank-3 structure of the matrix. The column on the extreme right of Table 1 indicates the average percentage of pixels in each image (averaged over all images) that were corrupted by shadows and specularities, respectively. We note that even when more than 30% of the pixels are corrupted by shadows and specularities, our method can efficiently retrieve the surface normals.

b. Textured scene. We also test our method using a textured scene. Like the traditional photometric stereo approach, our method does not have a dependency on the albedo distribution and works well on such scenes.

We use 40 images of Caesar for this experiment with each image generated under a different lighting condition (see Fig. 1(d) for example input image). The estimated normal maps as well as the pixel-wise error maps are shown in Fig. 3. We provide a quantitative comparison in Table 2 with respect to the ground-



(a) Ground truth (b) Our method (c) Least Squares (d) Error map (our method) (e) Error map (LS)

Fig. 3. Textured scene with specularity. 40 different images of Caesar were generated with texture, using the Cook-Torrance model for specularities. (a) Ground truth normal map with reference sphere. (b) and (c) show the surface normals recovered by our method and LS, respectively. (d) and (e) show the pixel-wise angular error w.r.t. the ground truth.

Table 2. Textured scene with specularity: Statistics of angle errors. We use 40 images under different illuminations.

Object	Mean error (in degrees)		Max error (in degrees)	
	LS	Our method	LS	Our method
Caesar	2.4	**0.016**	32.2	**0.24**

truth normal map. From the mean and maximum angular errors, it is evident that our method performs much better than the LS approach in this scenario.

c. Effect of the number of input images. In the above experiments, we have used images of the object under 40 different illuminations. In this experiment, we study the effect of the number of illuminations used. In particular, we would like to find out empirically the minimum number of images required for our method to be effective. For this experiment, we generate images of Caesar using the Cook-Torrance reflectance model, where the lighting directions are generated at random. The mean percentage of specular pixels in the input images is maintained approximately constant at 10%. The angular difference between the estimated normal map and the ground truth is used as a measure of accuracy of the estimate.

We present the experimental results in Table 3. We observe that with 5 input illuminations, estimates of both algorithms are very inaccurate but our method is worse than LS. However, when the number of illuminations is larger than 10, we observe that the mean error in the LS estimate becomes higher than that our method. Upon increasing the number of images further, the proposed method consistently outperforms the LS approach. If the number of input images is less than 20, then the maximum error in the LS estimate is smaller than that of our method. However, our method performs much better when at least 25 different illuminations are available. Thus, the proposed technique performs significantly better as the number of input images increases.

Table 3. Effect of number of input images. We use synthetic images of Caesar under different lighting conditions. The number of illuminations is varied from 5 to 40. The angle error is measured with respect to the ground truth normal map. The illuminations are chosen at random, and the error has been averaged over 20 different sets of illumination.

Num of images		5	10	15	20	25	30	35	40
Mean error	LS	**4.5**	0.52	0.51	0.53	0.62	0.59	0.59	0.57
(in degrees)	Our method	15.1	**0.23**	**0.036**	**0.026**	**0.015**	**0.019**	**0.017**	**0.013**
Max. error	LS	**88.2**	**34.5**	**13.7**	9.0	8.4	7.6	7.6	7.0
(in degrees)	Our method	127.9	56.6	25.6	**5.8**	**0.42**	**0.48**	**0.37**	**0.37**

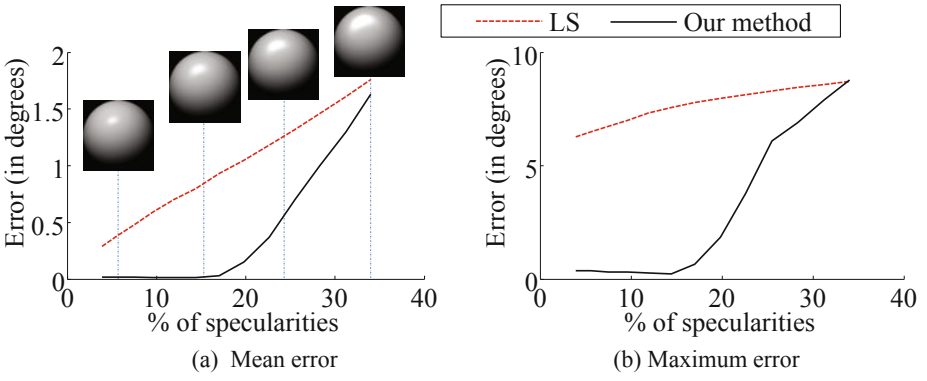


Fig. 4. Effect of increasing size of specular lobes. We use synthetic images of Caesar under 40 randomly chosen lighting conditions. (a) Mean angular error, (b) Maximum angular error w.r.t. the ground truth. The illuminations are chosen at random, and the error has been averaged over 10 different sets of illumination. (a) contains illustrations of increasing size of specular lobe.

d. Varying amount of specularity. From the above experiments, it is clear that the proposed technique is quite robust to specularities in the input images when compared to the LS method. In this experiment, we empirically determine the maximum amount of specularity that can be handled by our method. We use the Caesar scene under 40 randomly chosen illumination conditions for this experiment. On an average, about 20% of the pixels in each image is corrupted by attached shadows. We vary the size of the specular lobe in the input images (as illustrated in Fig. 4(a)), thereby varying the number of corrupted pixels. We compare the accuracy of our method against the LS technique using the angular error of the estimates with respect to the ground-truth.

The experimental results are illustrated in Fig. 4. We observe that our method is very robust when up to 16% of all pixels in the input images are corrupted by specularities. The LS method, on the other hand, is extremely sensitive to

Table 4. Handling more specularities by appropriately choosing λ. We use 40 images of Caesar under different lighting conditions with about 28% specularities and 20% shadows, and set $\lambda = C/\sqrt{m}$.

C	1.0	0.8	0.6	0.4
Mean error (in degrees)	1.42	0.78	0.19	0.029
Max. error (in degrees)	8.78	8.15	1.86	0.91

even small amounts of specularities in the input images. The angular error in the estimates of both methods rises as the size of the specular lobe increases.

e. Enhancing performance by better choice of λ. We recall that λ is a weighting parameter in our formulation given by Eq. 10. In all the above experiments, we have fixed the value of the parameter $\lambda = 1/\sqrt{m}$, as suggested by [14]. While this choice promises a certain degree of error correction, it may be possible to correct larger amounts of corruption by choosing λ appropriately, as demonstrated in [26] for instance. Unfortunately, the best choice of λ depends on the input images, and cannot be determined analytically.

We demonstrate the effect of the weighting parameter λ on a set of 40 images of Caesar used in the previous experiments. In this set of images, approximately 20% of the pixels are corrupted by attached shadows and about 28% by specularities. We choose $\lambda = C/\sqrt{m}$, and vary the value of C. We evaluate the results using angular error with respect to the ground-truth normal map. We observe from Table 4 that the choice of C influences the accuracy of the estimated normal map. For real-world applications, where the data is typically noisy, the choice of λ could play an important role in the efficacy of our method.

f. Computation. The core computation of our method is solving a convex program Eq. 10. For the specular Caesar data (Fig. 1(b)) with 40 images of 450×350 resolution, a single-core MATLAB implementation of our method takes about 7 minutes on a Macbook Pro with a 2.8 *GHz* Core 2 Duo processor and 4 GB memory, as against 42 seconds taken by the LS approach. While our method is slower than the LS approach, it is much more accurate in a wide variety of scenarios and is more efficient than other existing methods (*e.g.* [10]).

4.2 Qualitative Evaluation with Real Images

We now test our algorithm on real images. We use a set of 40 images of a toy Doraemon and Two-face taken under different lighting conditions (see Fig. 5(a), (d)). A glossy sphere was placed in the scene for light source calibration when capturing the data. We used a Canon 5D camera with the RAW image mode without Gamma correction. These images present new challenges to our algorithm. In addition to shadows and specularities, there is potentially additional noise inherent to the acquisition process as well as possible deviations from the idealistic Lambertian model illuminated by distant lights. In this experiment,

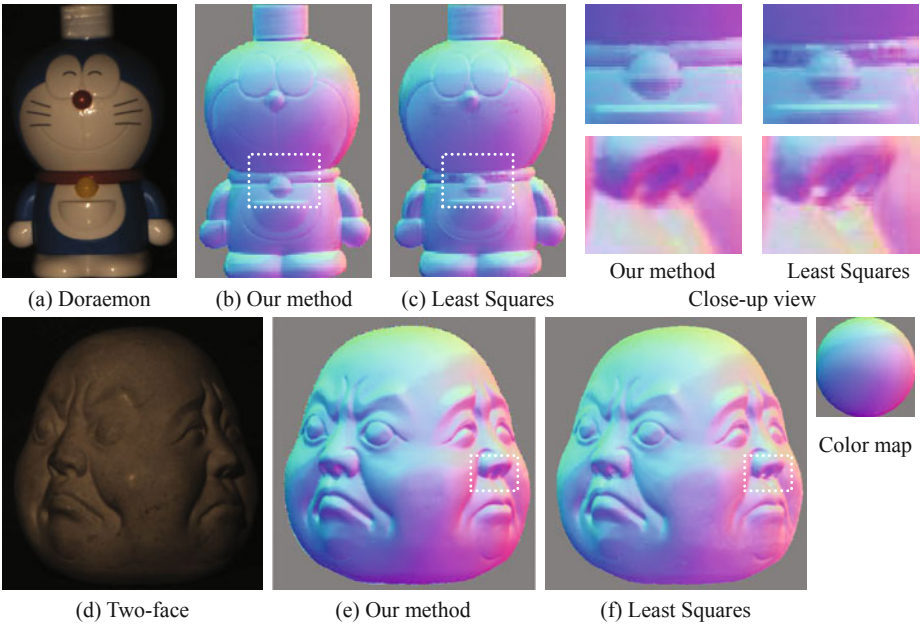


Fig. 5. Qualitative comparison on real data. We use images of Doraemon and Two-face taken under 40 different lighting conditions to qualitatively evaluate the performance of our algorithm against the LS approach. (a),(d) Sample input images. (b),(e) Normal map estimated by our method. (c),(f) Normal map estimated by Least Squares. Close-up views of the dotted rectangular areas (top-right) where the normal map estimate of our method is much smoother and realistic than that of Least Squares.

we use a threshold of 0.01 to detect shadows in images.[4] We also found experimentally that setting $\lambda = 0.3/\sqrt{m}$ works well for these datasets.

Since the ground truth normal map is not available for these scenes, we compare our method and the LS approach by visual inspection of the output normal maps shown in Fig. 5(b),(c),(e),(f). We observe that the normal map estimated by our method appears smoother and hence, more realistic. This can be observed particularly around the necklace area in Doraemon and nose area in Two-face (see Fig. 5) where the LS estimate exhibits some discontinuity in the normal map.

5 Discussion and Future Work

In this paper, we have presented a new computational framework to aid in photometric stereo. We have formulated the basic photometric stereo problem as a convex optimization problem that can be solved efficiently. The efficacy of our method is demonstrated using synthetic and real images. The biggest advantage of the proposed technique is its ability to handle shadows, specularities, and other kinds of large-magnitude, non-Gaussian errors in the data.

[4] All pixels are normalized to have intensity between 0 and 1.

The new framework also opens up several avenues for future research. Currently, we assume that all the images are noise-free and perfectly aligned with each other at the pixel level. However, in real world scenarios, small noise and misalignment are commonplace in any data acquisition process. By exploring the low-rank structure described in this work, we believe that the proposed technique can be extended to simultaneously handle small noise and misalignment in the input images.

Acknowledgment. This work was supported by grants ONR N00014-09-1-0230, NSF CCF 09-64215, and NSF ECCS 07-01676. Boxin Shi will be moving to the University of Tokyo in Fall 2010.

References

1. Woodham, R.: Photometric method for determining surface orientation from multiple images. Optical Engineering 19, 139–144 (1980)
2. Silver, W.: Determining shape and reflectance using multiple images. Master's thesis, MIT (1980)
3. Shashua, A.: Geometry and photometry in 3d visual recognition. Ph.D dissertation, Department of Brain and Cognitive Science, MIT (1992)
4. Belhumeur, P., Kriegman, D.: What is the set of images of an object under all possible lighting conditions? In: Proc. of CVPR, pp. 270–277 (1996)
5. Basri, R., Jacobs, D.: Lambertian reflectance and linear subspaces. PAMI 25, 218–233 (2003)
6. Georghiades, A.S., Kriegman, D.J., Belhumeur, P.N.: From few to many: illumination cone models for face recognition under variable lighting and pose. PAMI 23, 643–660 (2001)
7. Jolliffe, I.: Principal Component Analysis. Springer, Heidelberg (1986)
8. Fischler, M., Bolles, R.: Random sample consensus: A paradigm for model-fitting with applications to image analysis and automated cartography. Communications of the ACM 24, 381–395 (1981)
9. Hernández, C., Vogiatzis, G., Cipolla, R.: Multi-view photometric stereo. PAMI 30, 548–554 (2008)
10. Miyazaki, D., Hara, K., Ikeuchi, K.: Median photometric stereo as applied to the segonko tumulus and museum objects. IJCV 86, 229–242 (2010)
11. Mukaigawa, Y., Ishii, Y., Shakunaga, T.: Analysis of photometric factors based on photometric linearization. JOSA 24, 3326–3334 (2007)
12. Hayakawa, H.: Photometric stereo under a light source with arbitrary motion. JOSA 11, 3079–3089 (1994)
13. Knill, D.C., Mamassian, P., Kersten, D.: The geometry of shadows. JOSA 14, 3216–3232 (1997)
14. Wright, J., Ganesh, A., Rao, S., Peng, Y., Ma, Y.: Robust principal component analysis: Exact recovery of corrupted low-rank matrices by convex optimization. In: Proc. of Neural Information Processing Systems (2009)
15. Chandrasekaran, V., Sanghavi, S., Parrilo, P.A., Willsky, A.S.: Sparse and low-rank matrix decompositions. In: Proc. of IFAC Symp. on System Identification (2009)
16. Lin, Z., Chen, M., Wu, L., Ma, Y.: The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices (2009) (preprint)

17. Recht, B., Fazel, M., Parillo, P.: Guaranteed minimum rank solution of matrix equations via nuclear norm minimization. SIAM Review (2008) (to appear)
18. Candès, E., Recht, B.: Exact matrix completion via convex optimzation. Found. of Comput. Math. (2008)
19. Candès, E., Tao, T.: The power of convex relaxation: Near-optimal matrix completion. IEEE Transactions on Information Theory (2009) (to appear)
20. Ganesh, A., Lin, Z., Wright, J., Wu, L., Chen, M., Ma, Y.: Fast algorithms for recovering a corrupted low-rank matrix. In: Proc. of CAMSAP (2009)
21. Toh, K., Yun, S.: An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. Pacific Journal of Optimization (2009) (accepted)
22. Bertsekas, D.P.: Nonlinear Programming. Athena Scientific, Belmont (2004)
23. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problem. SIAM Journal on Imaging Sciences, 183–202 (2008)
24. http://www-roc.inria.fr/gamma/gamma/download/download.php: 3D meshes research database by INRIA gamma group (2008)
25. Cook, R.L., Torrance, K.E.: A reflectance model for computer graphics. SIGGRAPH Comput. Graph. 15, 307–316 (1981)
26. Ganesh, A., Wright, J., Li, X., Candès, E., Ma, Y.: Dense error correction for low-rank matrices via principal component pursuit. In: Proc. of ISIT (2010)

Robust Auxiliary Particle Filter with an Adaptive Appearance Model for Visual Tracking

Du Yong Kim, Ehwa Yang, Moongu Jeon, and Vladimir Shin

School of Information and Mechatronics,
Gwangju Institute of Science and Technology
{duyong,ehwa,mgjeon,vishin}@gist.ac.kr

Abstract. The algorithm proposed in this paper is designed to solve two challenging issues in visual tracking: uncertainty in a dynamic motion model and severe object appearance change. To avoid filter drift due to inaccuracies in a dynamic motion model, a sliding window approach is applied to particle filtering by considering a recent set of observations with which internal auxiliary estimates are sequentially calculated, so that the level of uncertainty in the motion model is significantly reduced. With a new auxiliary particle filter, abrupt movements can be effectively handled with a light computational load. Another challenge, severe object appearance change, is adaptively overcome via a modified principal component analysis. By utilizing a recent set of observations, the spatiotemporal piecewise linear subspace of an appearance manifold is incrementally approximated. In addition, distraction in the filtering results is alleviated by using a layered sampling strategy to efficiently determine the best fit particle in the high-dimensional state space. Compared to existing algorithms, the proposed algorithm produces successful results, especially when difficulties are combined.

1 Introduction

Problems associated with visual tracking have been heavily investigated in the computer vision community. Although a number of algorithms were suggested from this research community, challenging issues still remain when they are implemented in real-world circumstances. These challenges mainly stem from two primary sources: uncertainty in a dynamic model, and design issues related to development of an accurate observation likelihood function. Among the many types of tracking frameworks, we focus on appearance-based tracking because it is widely used and is more dependent on these issues than other approaches.

In conventional particle filtering methods for visual tracking, a set of particles is drawn from the importance density (state transition density in conventional particle filtering), which is the distribution of predicted locations of an object. If the predicted particles do not include the true position or the best fit location is not accurately determined, filter distractions will occur and then gradually adapt to the non-targets, resulting in filter drift. Thus, the main source of filter drift is a combination of inaccurate predictions and an incorrectly cropped target appearance. These two things are involved with the uncertainty in given dynamic motion model and approximation error of observation likelihood function.

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 718–731, 2011.

We can intuitively overcome these problems by exploring a large search space with a sufficient number of samples, though this method requires heuristics and a high computational cost. Furthermore, the filtering accuracy is often not guaranteed. In previous research, dynamic motion model learning [1], optimal importance function (OIF) [2], and discriminative observation system design approaches [3], [4] have been suggested independently, and also combined in an attempt to solve this problem. As noted in previous works [5], [6] the performance of the adaptive appearance model is directly related to the hypothesis parameter estimation of the object appearance. When the filter starts experiencing effects of distraction caused by inaccurate parameter estimation due to poor predictions, it is difficult to recover the correct trajectory because the distraction critically affects update of the adaptive appearance model. Because the accuracy of the suggested dynamic motion model and updates of the reference appearance are closely related, these two problems are dependent and need to be considered simultaneously. In order to solve this 'chicken and egg' problem, numerous attempts have been made to design an adaptive appearance model [7], [8], [9], [5]. However, even if the appearance model behaves moderately well, the advantages can be negated by inaccurate predictions.

In this paper, we propose a novel approach for simultaneous adaptive appearance learning and robust filtering. The main challenges in visual tracking mentioned above are resolved by the following two main contributions of our proposed algorithm. 1) For more accurate and robust appearance learning, an adaptive incremental principal component analysis (PCA) is devised. The adaptive incremental PCA provides enough flexibility to cover a wide range of different situations including abrupt changes in pose, sudden changes in illumination, and fast motion, without having to reset the parameters. 2) To effectively utilize the new adaptive appearance model and obtain correct updates, the level of uncertainty in the learned dynamics is reduced by using the receding horizon estimation (RHE) method in a particle filtering framework. By utilizing a temporal set of observations, RHE achieves robustness against uncertainty in dynamic models by internal iterations of receding windows [10], [11]. Additionally, filter distraction is avoided by using a layered sampling method to determine the most effective sample in a high-dimensional state space.

The remainder of this paper is organized as follows. In the next section, an overview of research related to robust visual tracking is provided. A newly proposed filter using RHE, adaptive appearance learning and the robust observation function design are then introduced in Section 3 as the main framework of the proposed algorithm. In Section 4, several challenging sequences are tested to verify the usefulness of the proposed work over state-of-the-art algorithms. Finally, conclusions are provided in Section 5.

2 Related Works

Many recent works in the appearance-model-based tracking have focused on the design of a robust filter and an adaptive appearance model. However, they faced the main challenge when abrupt motions and severe appearance changes should

be simultaneously dealt with. To this end, the adaptive appearance model should not be considered independently in a template update-based tracking, because the adaptive appearance model update is closely related with hypothesis parameter (e.g., center position and scale) estimation. Therefore, reliable parameter estimation and the adaptive appearance model should be intelligently combined in a unified framework.

First, the adaptive appearance model is viewed as a key design issue because filter drift problems arise when abrupt motion or illumination changes are not well reflected in the reference appearance. To tackle such difficulties, several adaptive appearance modeling methods have been suggested in the literatures as follows.

In order to update appearance changes, a pixel-wise representation was sequentially estimated using a kernel-based Bayesian filter [7]. In this filter, the Gaussian mixture representation was introduced as a basic framework for a compact and adaptive representation in order to handle multi-modal intensity of appearance. However, this algorithm tends to suffer from high computational complexity because a single target is regarded as a patch for a multi-modal distribution of pixels. Additionally, appearance is updated in every frame that is not inherently robust.

Instead of representing the reference template as a pixel-wise multi-modal distribution, a more compact and efficient approach has been suggested based on the incremental update [8], [5], [12]. In this case, object appearance is considered as a temporal subspace in a manifold, and then PCA is used to approximate the linear subspace of the manifold. Manifold learning methods have also been proposed in the appearance-based tracking to handle pose and illumination changes of appearance [13], [12], however, this method requires a training set that contains sufficient samples to cover all possible outcomes. Moreover, obtaining the transition probability between linear subspaces is not an easy task.

Recent PCA-based appearance learning method is proposed based on the online learning, i.e., it does not require the offline learning with a training set [8]. This approach temporarily learns the linear subspace via PCA modified to support a sequential update framework. The proposed incremental version of PCA enables the tracker to capture the local temporal coherence of the manifold, and does not need to construct the whole space. However, the incremental PCA-based subspace learning proposed in [8] is only suitable for gradual changes in appearance, as that it cannot deal with abrupt pose and/or illumination changes.

Second, the other challenge in visual tracking is to avoid filter failure due to an inaccurate state dynamic model. In particle filtering for visual tracking, random walk models have been widely adopted because exact dynamic modeling of an object is nearly impossible. Usually, however, random walk models often fail to correctly predict the expected position of an object when an abrupt motion occurs. To avoid uncertainty of a dynamic motion model, a combination of auto-regressive (AR) dynamic motion model learning [14] and the OIF method was proposed for geometric particle filtering in affine group [2]. According to research, the combined AR dynamic model and OIF were able to significantly improve the performance compared to the original incremental learning based visual tracker [8] in terms

of handling sudden motions. Because OIF reflects a current observation, it was able to efficiently avoid filter drift from abrupt motions or a low frame rate video. In OIF-based visual tracking, however, the need for complex computations in the image warping process and the analytic calculation of optimal importance function for every particle makes real-time implementation impossible. Furthermore, there is no consideration of adaptive appearance learning for severe appearance changes and sudden pose variations. Especially when sudden pose changes occur Jacobian-based optimal importance function is not useful anymore because it is not analytically differentiable. The situation becomes catastrophic when the object sudden moves are combined with severe change in appearance.

3 Receding Horizon Auxiliary Particle Filter

The primary purpose of RHE is to achieve robustness against uncertainty in a dynamic model. In the control and estimation fields, RHE has already been successfully implemented for the rectilinear orbit problem [10], estimating aircraft engine temperature [11], and other models. To the best of our knowledge, however, RHE has never been applied to appearance-based robust visual tracking under a particle filtering framework.

3.1 Particle Filter for Appearance Based Visual Tracking

Let the state vector $x_t \in \Re^6$ represent components in the local coordinate based approach, in other words, the x- and; y-position of the box center $p_t = (p_{x,t}, p_{y,t})$, the scale of the box $S_t = (S_{x,t}, S_{y,t})$, the rotation angle, ϕ_t, and the skew direction θ_t of an object as described in Fig. 1.

Then, the aim of probabilistic tracking is to estimate x_t based on the probability density function of x_t given through the observation set $z_{1:t} = \{z_1, ..., z_t\}$ and described as two Bayesian recursion equations:

$$Prediction : p(x_{t+1}|z_{1:t}) = \int p(x_{t+1}|x_t)p(x_t|z_{1:t})dx_t, \tag{1}$$

$$Update : p(x_{t+1}|z_{1:t+1}) \propto p(z_{t+1}|x_{t+1})p(x_{t+1}|z_{1:t}). \tag{2}$$

In the prediction step for the conventional particle filter, a set of bounding boxes(samples) are drawn from the state transition density $p(x_{t+1}|x_t)$.

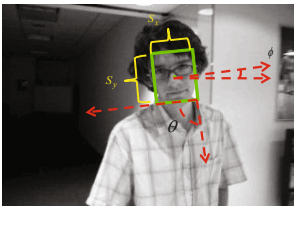


Fig. 1. Local coordinate based appearance model

The observation likelihood $p(z_{t+1}|x_{t+1})$ is then evaluated via the probability distribution of image patches with respect to the reference appearance of the target. Note that the observation process in the appearance-based visual tracking is the warping of image patches as illustrated in the Fig. 1.

Because motion of a target is often unpredictable, and the observation system is usually not explicitly described, the CONDENSATION algorithm (another name of particle filter) [15] has been extensively used to implement Bayesian recursions for visual tracking applications. Even though it is widely used due to its flexibility, improvements are necessary to achieve robustness and accuracy.

Particularly, in the appearance-based tracking, the CONDENSATION algorithm often suffers from loss of tracks caused by inaccurate predictions and approximation errors in observation likelihood due to the high-dimensional state space and issues related to adaptive appearance learning. The state-of-the-art implementation of the CONDENSATION algorithm in the appearance-based tracking is known as the incremental visual tracker (IVT), which learns the appearance using incremental PCA approach [8]. In this paper, IVT is taken as a main reference to explain the body of our work and comparisons.

3.2 Receding Horizon Auxiliary Particle Filter

The CONDENSATION algorithm often fails in visual tracking when the observation likelihood density is peak and overlapped near the tail of the state transition density. In such a case, samples in high-likelihood regions are ignored due to low probabilities of predicted samples.

To overcome the problem, efficient design of the importance density was proposed in the literatures [16], [17]. However, when there is uncertainty in the dynamic motion model, it is not effective anymore. Heuristics with a large numbers of particles may work, however, it requires extremely intense computation makes it not useful in real-time applications and accurate performance is not guaranteed.

RHE is an adaptive estimation strategy in which the most recent finite set of observations is collected and processed within the receding window. The main concept of RHE is illustrated in Fig. 2. Given the initial condition, $p(x_{t-\Delta}|z_{1:t-\Delta})$, the receding horizon filter recursively calculates intermediate estimates inside the window until it obtains a final estimate as the estimate of the current time stamp, where Δ is the window length. In linear dynamic systems RHE has been proven to be robust when there is unknown modeling uncertainty in the system [10], [11]. To realize the RHE in the particle filtering framework for visual tracking, we propose the multi-staged filtering approach based on the auxiliary particle filter (APF).

Instead of using the state transition density $p(x_{t+1}|x_t)$ as the importance density, a set of most recent observations is incorporated to design the more efficient importance density. By doing so, inter-frame information of the state is reflected in the importance density because a set of particles is moving near the filtered estimates. A newly designed importance density of the proposed method is defined as

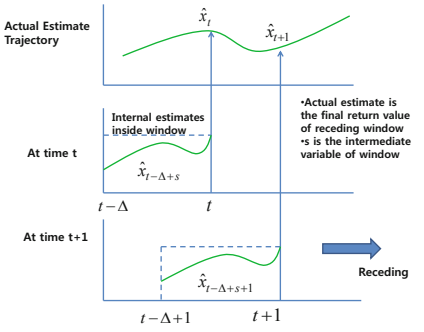


Fig. 2. Receding horizon estimation strategy

$$q(x_{t+1}^i|x_t^i, z_{1:t+1}) \approx q(x_{t+1}^i|x_t^i, z_{t-\Delta:t+1}) = p(x_{t+1}^i|x_t^i)p(z_{t+1}|\hat{x}_t), \tag{3}$$

where i is the particle index, $\hat{x}_t$ is the auxiliary estimate given as $\hat{x}_t = \text{argmax}_{x_t} p(x_t|z_{t-\Delta:t})$. Here, $\hat{x}_t$ is obtained by sequentially calculating the intermediate estimates from $s = t - \Delta$ up to time $s = t$ and it is implemented via auxiliary particle filtering method [18]. Additionally, to adaptively reduce uncertainty in the dynamic motion model, layered sampling-based state transition density is used given based on the Gaussian density as

$$p(x_{t+1}|x_t^i) = \mathcal{N}(\hat{x}_t, \beta^{\Delta}\Psi), \tag{4}$$

where the diagonal covariance matrix, Ψ, whose elements are corresponding variances for each parameter is initially defined with sufficiently large value enough to cover abrupt motion. Then, in the intermediate iteration of RHE window, the layered sampling strategy is done sequentially by multiplying the annealing parameter $\beta^m \in \Re$ where $1 \geq \beta^1 ... > \beta^{\Delta} > 0$, and $m = 1, ..., \Delta$. In the implementation of the algorithm, we use a Gaussian random walk model of annealed state transition density as (4) instead of the learned AR dynamic model. As such the number of stages is equal to the window length to incrementally reduce the search space.

The APF proposed by Pitt el. al. attempted to reflect the current observation in the importance sampling so that APF outperforms the conventional particle filter (PF) in many applications [18], [19]. However, it was pointed out that APF may only achieve little improvement in visual tracking applications because it may also show worse performance if the dynamic model $p(x_{t+1}|x_t)$ is quite scattered or if the likelihood varies significantly over the prior $p(x_{t+1}|x_t)$.

Under the APF framework, we apply the RHE concept to overcome uncertainty in the dynamic model without employing OIF, which requires substantially additional computational resources. OIF approach requires the analytic importance density calculation with Jacobians for each particle that is computationally intense; whereas our approach only needs observation function

evaluation. Using a new importance density given in (3), particle weights in receding widow are calculated as:

$$\omega_{s+1}^{i} = \frac{p(x_{s+1}^{i}|x_{s}^{i})p(z_{s+1}|x_{s+1}^{i})}{q(x_{s+1}^{i}|x_{s}^{i}, z_{s+1})} = \frac{p(x_{s+1}^{i}|x_{s}^{i})p(z_{s+1}|x_{s+1}^{i})}{p(x_{s+1}^{i}|x_{s}^{i})p(z_{s+1}|\hat{x}_{s})} = \frac{p(z_{s+1}|x_{s+1}^{i})}{p(z_{s+1}|\hat{x}_{s})}. \quad (5)$$

Initially, relatively a small number of samples are drawn over the large search space area. Then, as the sliding window proceeds variance of state transition density is decreased and a sequence of recent observations is utilized to relocate particles around the high likelihood region for efficient sampling. For the initial condition of the receding window, we suggest using the previously obtained estimate of the filter at time $t - \Delta$. The summary of proposed algorithm, Receding horizon auxiliary particle filter (RHAPF) is given in Algorithm 1.

Algorithm 1. Receding horizon auxiliary particle filter (RHAPF)

for $t = 1, 2, ...$
  if $t \leq \Delta$,Run CONDENSATION tracker
  $p(x_t|z_t) \approx \sum_{i=1}^{N} \frac{1}{N}\delta(x_t - x_t^i), \hat{x}_t = \frac{1}{N}\sum_{i=1}^{N} x_t^i$, where N : number of samples
  else
  Initialize the sliding window with $\left\{x_{t-\Delta}^{i}\right\}_{i=1}^{N}$, where $x_{t-\Delta}^{i} \sim \mathcal{N}(\hat{x}_{t-\Delta}, \Psi)$
  for $s = t - \Delta, ..., t$
    for $i = 1, ..., N$
      1. Set $p(x_{s+1}|x_s^i) = \mathcal{N}(\hat{x}_s, \beta^m \Psi), m = s - t + \Delta$
      : annealed state transition density
      2. Draw $x_{s+1}^i \sim p(x_{s+1}|x_s^i)$
      3. Update weight using (5)
      4. Normalize weights so that $\sum_{i=1}^{n} \omega_{s+1}^i = 1$
    end for
  Calculate auxiliary estimates $\hat{x}_{s+1} = \text{argmax}_{x_{s+1}^i}(\omega_{s+1}^i)$
  Resampling step from [15]
  end for
  $p(x_t|z_t) \approx \sum_{i=1}^{N} \frac{1}{N}\delta(x_t - x_{s=t}^i), \hat{x}_t = \text{argmax}_{x_{s=t}^i}(\omega_{s=t}^i)$
  end if
end for

At a glance, RHAPF may have the same amount of or more computations than the one shot particle filter with a sufficient number of particles over a large space. However, the proposed importance density intelligently adapts the unexpected dynamics by multi-staged sequential filtering with relatively small numbers of particles. Note that the proposed algorithm resembles the mean-shift type algorithm [7]. However, our algorithm uses stochastic search that can be understood as a combination of layered sampling [20] and auxiliary particle filter [18] under the RHE framework. For the successful design of visual tracker in unconstrained situations, adaptive appearance model learning and accurate observation likelihood calculation should be combined with RHAPF. To this extend, we propose an adaptive appearance model learning method for accurate approximation of the observation

likelihood function not to adapt non-targets that will be detailed discussed in the next section. Consequently, including adaptive appearance model learning and an observation function approximation using layered sampling, the implementation of RHAPF for visual tracking application is provided.

3.3 Adaptive Appearance Learning and Robust Observation Function

In computer vision applications, dealing with the high dimensionality of state vectors and accurately calculating the observation likelihood are very important but difficult issues in observation system design. In the local coordinate based appearance model, we have a six-dimensional state vector; therefore, it is almost impossible to construct a true observation likelihood distribution. To deal with the high dimensionality of appearance, we adopt the incremental PCA subspace learning method [8] for template learning. In IVT [8], the observation system is expressed as

$$z_t = h(I(w(\chi; x_t))) + v_t, \tag{6}$$

where χ denotes the pixel corrdinate, and $w(\chi; x_t)$ describes the warping function of the given image $I(\chi)$ according to the state $x_t = (p_{x,t}, p_{y,t}, S_{x,t}, S_{y,t}, \phi_t, \theta_t)$ defined in Section 3.1.

In the incremental PCA based observation function, we calculate the mean and principal eigenvectors and incrementally update them for the reference template appearance. As such, if we let $\bar{T}(\chi; x_t)$ and $g_j(\chi; x_t)$, $j = 1, ..., L$, denote the template mean and L principal eigenvectors, we can represent the reconstruction error vector for $I(w(\chi; x_t)$ considering matrices as the stacked vector forms

$$e^2 = \|I(w(\chi; x_t)) - \sum_j c_j g_j(\chi; x_t)\|^2, \tag{7}$$

where $c_j = \sum_\chi g_j(\chi; x_t)(I(w(\chi; x_t)) - \bar{T}(\chi; x_t))$ are the coefficients from the projection of the template mean to each principal eigenvector $g_j(\chi; x_t)$.

The subspace learning method used in [8] updates the subspace of object appearance at every fifth frame by incrementally calculating the eigenvectors and the mean. However, even if the method applies the forgetting factor inside the update process to down-weight the effect of old observations, it is still not sufficient for handling drastic changes in appearance. Furthermore, as pointed out in [21], fragility of the object representation can degrade over a long image sequence. To alleviate such disadvantages, we designed a sliding window-based PCA subspace learning method. In our proposed method, instead of applying the forgetting factor to reduce the effect of old observations, we suggest that every window be initialized by using the previously estimated template mean and eigenvectors. By doing so, no tuning of the forgetting factor is required.

For the simplicity, let us denote $\bar{T}_t$, S_t, and I_t as the template mean, scatter matrix and warped image at t. The purpose of the proposed algorithm is to update $\bar{T}_t$ and S_t based on the initial conditions $\left\{\bar{T}_{t-\Delta}, S_{t-\Delta}\right\}$ and a recent set of observation $[I_{t-\Delta+1}, I_{t-\Delta+1}, ..., I_t]$. Here, we assume that the initial conditions

$\{\bar{T}_{t-\Delta}, S_{t-\Delta}\}$ summarize the previous observation set $[I_1, I_2, ..., I_{t-\Delta}]$. Then, we recursively update $\{\bar{T}_{t+1}, S_{t+1}\}$ with the initial conditions $\{\bar{T}_{t-\Delta+1}, S_{t-\Delta+1}\}$ and a new recent observation set $[I_{t-\Delta+2}, I_{t-\Delta+3}, ..., I_{t+1}]$. Therefore, when the window is receding, the oldest observation is automatically discarded and the new one is included.

In Algorithm 2, the modified version of the incremental PCA subspace learning method considering a sliding window is described in order to obtain the updated mean and the scatter matrix for the eigenvectors. Here, we adopt the incremental update algorithm for the mean and the scatter matrix used in [22]. The proposed algorithm is inspired by the premise that the nonlinear manifold of object appearance can be decomposed via piecewise linear subspace manifolds [23]. The proposed PCA based sliding window subspace learning does not approximate the structure of object manifold, instead it attempts to learn the spatio-temporal subspace as accurately and adaptively as possible.

Algorithm 2. Sliding window based adaptive incremental PCA

Given initial conditions $\{\bar{T}_{t-\Delta}, S_{t-\Delta}\}$ and $[I_{t-\Delta+1}, I_{t-\Delta+2}, ..., I_t]$,
Calculate $\bar{T}_\Delta = \frac{1}{\Delta}\sum_{i=t-\Delta+1}^{t}$, $S_\Delta = \sum_{i=t-\Delta+1}^{t}(I_i - \bar{T}_\Delta)(I_i - \bar{T}_\Delta)^T$.
Sets of template mean and the scatter matrix: $\{\bar{T}_{t-\Delta}, S_{t-\Delta}\}$ and $\{\bar{T}_\Delta, S_\Delta\}$
Update mean and scatter matrix $\bar{T}_t$ and S_t as
$\bar{T}_t = (\bar{T}_{t-\Delta} + \bar{T}_\Delta)/2$, $S_t = \frac{\Delta^2}{\Delta}(\bar{T}_{t-\Delta})(\bar{T}_{t-\Delta})^T + S_{t-\Delta} + S_\Delta$
Consider new initial conditions and a new set
$\{\bar{T}_{t-\Delta+1}, S_{t-\Delta+1}\}$ and $[I_{t-\Delta+2}, I_{t-\Delta+3}, ..., I_{t+1}]$ for next step

Another important issue in the robust visual tracking is the filter distraction. The filter distraction is a challenging task in the high dimensional state space because it is very difficult to efficiently identify the high likelihood region. In the appearance-based tracking, if the true appearance is not precisely approximated in the observation system, the filter will gradually adapt to the non-target or the estimate becomes biased. Therefore, many researchers have attempted to

Fig. 3. Comparison of filter distraction and partial occlusion handling (red box: IVT, green box: proposed method via the layered samping, row 1: 'daivd' and row 2: 'dudek' sequences)

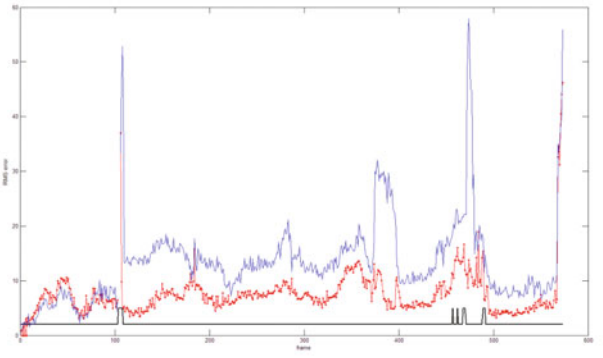


Fig. 4. RMSE and detected occlusion (blue: IVT, red: RHAPF, black: detected occlusion

design a robust and accurate observation function [24], [4], [20]. Based on the RHAPF and the sliding window-based adaptive incremental PCA, the proposed algorithm can efficiently identify the high-observation likelihood region and prevent the filter distraction by using annealed layered sampling.

To show the effectiveness of layered sampling based observation likelihood approximation, we compare with the original IVT as given in Fig. 3. As shown in Fig. 3, when an object suddenly changes its pose ('david' sequence), IVT temporarily experiences distraction, whereas the RHAPF with an adaptive template update method correctly adjusts to the change of appearance without distraction.Also, when the object is temporary occluded ('dudek' sequence), the proposed algorithm completely recovered without distraction whereas the IVT could not. For 'dudek' sequence, we provide the quantitative comparison results according to the root mean square error(RMSE) by pixel in Fig. 4. In the sequence, the object experiences occlusions, sudden motions with pose variations. In Fig. 4, the black solid line at the bottom indicating occlusion was detected in the sequence; it can be seen that our tracker simultaneously achieves improved accuracy and efficient occlusion handling. In summary, we provide the the overall algorithm by combining the RHAPF and adaptive appearance model learning through Algorithm 3.

Algorithm 3. Visual receding horizon auxiliary particle filter

for $t = 1, 2, ...$
  if $t \leq \Delta$, Run CONDENSATION tracker,
  Store template mean and eigenvectors
  else
    RHAPF iteration in Algorithm 1.
    Update reference template using Algorithm 2.
  end if
  $p(x_t|z_t) \approx \sum_{i=1}^{N} \frac{1}{N} \delta(x_t - x_{s=t}^i), \hat{x}_t = \text{argmax}_{x_{s=t}^i}(\omega_{s=t}^i)$
end for

4 Experiments

In this section, we test the robustness of RHAPF using several challenging sequences. The sequences contain the following difficulties: abrupt pose change, fast motion, temporal occlusion and severe illumination change. From comparative experiments the superiority of our tracker over the existing state-of-the art trackers can be verified.

IVT and the geometric PF with OIF are state-of-the-art algorithms based on the adaptive appearance model. The adaptive appearance model update was achieved using incremental PCA, however, as previously pointed out; the tracker suffers from temporal distraction and drift when abrupt pose and severe illumination changes are combined. Hence, designing a robust tracking algorithm in a unified framework to overcome these complicated situations is very important because such challenges do not occur separately in real-world circumstances. In these experiments, 'trellis' and 'sylvester' sequences were tested for abrupt movement and sudden illumination variation, which were not thoroughly tracked using IVT or the geometric PF with OIF. For all the experiments, the horizon length is set to as 2 3, the batch size for PCA is set as equal to the horizon length. The layered sampling is stopped when the certain threshold is met. For the fair comparisons, the same number of particles is used for different algorithms. Therefore, there was no tuning to adapt different situations.

Fig. 5. Comparison of tracking results (first row: IVT, second row: RHAPF)

4.1 Comparison with IVT

The most challenging sequence, 'trellis' is tested in Fig. 5. As shown in the figure, it basically has camera motions, lightning changes are severe and the object suddenly changes the pose simultaneously. IVT failed to track the object around t=340 because there is no adaptive appearance learning mechanism to handle the pose variations and illumination changes at once. In contrast, in the sequence around t=356, even though the object has abrupt motion with severe illuminations still our algorithm succeeded to track. From the experiment, visual RHAPF effectively tracks the object regardless of unconstrained situations. Here,

we do not include the result of the geometric PF with OIF, because it failed before IVT lost the track. That is because the geometric PF with OIF only handles the abrupt motions not appropriate for large pose changes.

4.2 Comparison with Geometric PF with OIF

The geometric particle filtering with an AR model learning approach [1] was improved with OIF to cope with abrupt motions thus shown to be more accurate than IVT [2]. However, as shown in Fig. 5, in the 'trellis' sequence results, when the geometric PF was implemented with OIF, even though the OIF approach has been proven somewhat effective, it has serious limitations and drawbacks when implemented in complicated situations. Due to the numerical instability of the local linearization method, it was not always applicable especially when temporal occlusion and large pose changes occurred. We additionally compared our tracker with the geometric PF with OIF through the 'sylvester' and 'toy' sequences, in which large pose changes are combined with abrupt movements (Fig. 6). In the 'sylvester' sequence, the tracking failure of IVT began at t=609, and the geometric PF with OIF began at t=1149. From the 'toy' sequence results, even without an optimal importance function, the performance of our tracker is better than that of the geometric PF with OIF with more accurate estimates. We also measured the computational time to show the superior efficiency of our tracker, which are provided in Table 1. The results clearly support that our tracker operates in near real time with remarkable performance compared to the geometric PF with OIF. In Table 1, IVT (conventional PF) is not considered because it always lost tracks. Note that these experiments were performed using an Intel Core-2 Duo 2.66 GHz processor and unoptimized Matlab software.

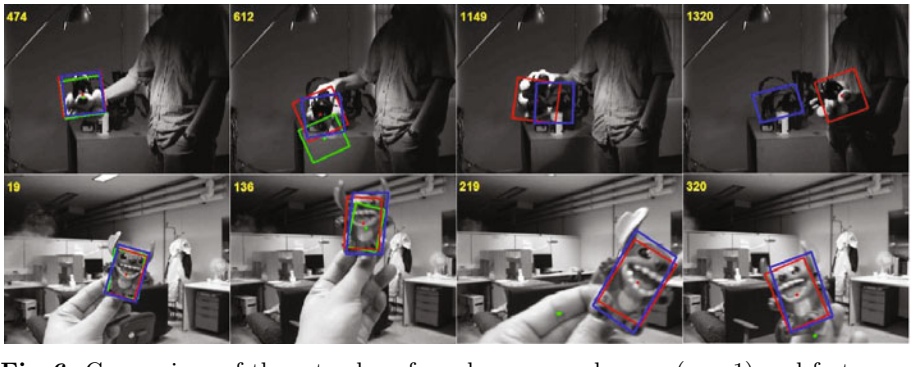


Fig. 6. Comparison of three trackers for a large pose changes (row 1) and fast movements (row 2) (green box: IVT, blue box: PF with OIF, red box: RHAPF)

Table 1. Measured computation time for 1 frame with 200 particles

Algorithm	Sylvester sequence	Toy sequence
Geometric PF with OIF	1.38s	1.35s
RHAPF	0.159s	0.130s

5 Conclusion

In this paper, we proposed a novel approach for handling several difficulties involved in visual tracking in a unified framework. The RHE strategy was proposed to effectively handle the fast movement of an object without using AR model learning or OIF in particle filtering. In terms of robust observation system design, the adaptive appearance learning method was suggested with a sliding window approach. The main advantage of our proposed method is that it resolves a combination of complex difficulties in a unified framework without needing to reset the parameters. This is a very attractive advantage because, in real-world situations, temporal occlusion, large pose changes, and abrupt movements are not likely to happen separately. Moreover, without employing an optimal importance function, currently a common solution for handling abrupt movements, RHAPF was able to correctly track an object without failure. As the simulation results illustrated, the proposed tracker required significantly fewer computations than the OIF approach, without loss of track even in complicated situations.

Acknowledgement. This work was supported in part by GIST Systems Biology Infrastructure Establishment Grant, and by the Center for Distributed Sensor Network at GIST.

References

1. Kwon, J., Park, F.C.: Visual tracking via particle filtering on the affine group. IJRR (2009)
2. Kwon, J., Lee, K.M., Park, F.C.: Visual tracking via geometric particle filtering on the affine group with optimal importance functions. In: CVPR (2009)
3. Barbenko, B., Yang, M.H., Belongie, S.: Visual tracking with online multiple instance learning. In: CVPR (2009)
4. Li, Y., Ai, H., Yamashida, T., Lao, S., Kawade, M.: Tracking in low fame rate video: A cascade particle filter with discriminative observers of different life spans. IEEE PAMI, 1728–1740 (2008)
5. Matthews, I., Ishikawa, T., Baker, S.: The template update problem. IEEE PAMI, 810–815 (2004)
6. Yang, M., Fan, Z., Wu, Y.: Tracking nonstationary visual appearances by data-driven adaptation. IEEE TIP, 1633–1644 (2009)
7. Han, B.H., Zhu, Y., Comaniciu, D., Davis, S.L.: Kernel-based bayesian filtering for object tracking. In: CVPR (2005)
8. Ross, R.D., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking. IJCV (2008)
9. Jepson, A.D., Fleet, J., Maraghi, T.F.: Robust online appearance models for visual tracking. IEEE PAMI (2001)
10. Jazwinski, A.: Stochastic process and filtering theory. Academic Press, London (1970)
11. Kwon, W.H., Kim, P.S., Park, P.G.: A receding horizon kalman fir filter for linear continuous-time systems. IEEE TAC (1999)
12. Lee, K.C., Ho, J., Yang, M.H., Kriegman, D.: Video-based face recognition using probabilistic appearance manifolds. In: CVPR (2003)

13. Lim, H., Morariu, V., Camps, O.I., Sznaier, M.: Dynamic appearance modeling for human tracking. In: CVPR (2006)
14. Blake, A., Isard, M.: Active contours. Springer, Heidelberg (1998)
15. Isard, M., Blake, A.: Contour tracking by stochastic propagation of conditional density. In: Buxton, B.F., Cipolla, R. (eds.) ECCV 1996. LNCS, vol. 1065. Springer, Heidelberg (1996)
16. Rui, Y., Chen, Y.: Online learning of probabilistic appearance manifolds for video-based recognition and tracking. In: CVPR (2001)
17. Doucet, A., Goodsil, S., Andrieu, C.: On sequential monte carlo sampling methods for bayesian filtering. Statistics and Computing, 197–208 (2000)
18. Pitt, M., Shephard, N.: Filtering via simulation: auxiliary particle filters. Journal of American Statistical Association, 590–599 (1999)
19. Vlassis, N., Terwijin, B., Kros, B.: Auxiliary particle filter robot localization from high-dimensional sensor observations. In: ICRA (2002)
20. Deutsher, J., Blake, A., Reid, I.: Articulated body motion capture by annealed particle filter. In: CVPR (2000)
21. Shen, C., Kim, J., Wang, H.: Generalized kernel-based visual tracking. IEEE CSVT, 119–130 (2010)
22. Levy, A., Lindenbaum, M.: Sequential karhunen-loeve basis extraction and its application to images. IEEE TIP, 1371–1374 (2000)
23. Lee, K.C., Kriegman, D.: Online learning of probabilistic appearance manifolds for video-based recognition and tracking. In: CVPR (2005)
24. Kim, M., Kumar, S., Pavlovic, V., Rowley, H.: Face tracking and recognition with visual constraints in real-world videos. In: CVPR (2008)

Sustained Observability for Salient Motion Detection

Viswanath Gopalakrishnan, Yiqun Hu, and Deepu Rajan

School of Computer Engineering
Nanyang Technological University, Singapore

Abstract. Detection of the motion of foreground objects on the backdrop of constantly changing and complex visuals has always been challenging. The motion of foreground objects, which is termed as salient motion, is marked by its predictability compared to the more complex unpredictable motion of the backgrounds like fluttering of leaves, ripples in water, smoke filled environments etc. We introduce a novel approach to detect this salient motion based on the control theory concept of 'observability' from the outputs, when the video sequence is represented as a linear dynamical system. The resulting algorithm is tested on a set of challenging sequences and compared to the state-of-the-art methods to showcase its superior performance on grounds of its computational efficiency and detection capability of the salient motion.

1 Introduction

Automated video analysis has been always looked upon as an important yet challenging task by the computer vision community. With the exponential growth of multimedia content including videos, produced both by professionals and amateurs, it has become imperative to develop intelligent methods to analyze and understand its content. The detection of salient motion in a video is quite important in this context as it can guide and help many other related tasks like object tracking and surveillance [19,16], video retargeting [12] etc. Surveillance and tracking applications will be aided by the information on 'what to follow' with the knowledge of salient motion, while video retargeting applications can use the knowledge of salient motion to adapt multimedia content with minimum distortion.

Identification of salient motion in a video is a challenging task, especially with dynamic backgrounds. Several methods have been proposed to solve the problem of salient motion detection. Background modeling is a prominent approach among them in which any unlikely event from the statistically modeled background will be regarded as salient motion [8,13]. Monnet et. al [1] used an online autoregressive model to predict the dynamic background and detection of salient motion is performed by comparing the prediction with actual observation. In [18], the continuously changing background is predicted using dynamic texture ARMA model and the foreground objects are detected as outliers to this model. The background modeling techniques require a learning process and this

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 732–743, 2011.

can be difficult in case foreground objects are present in the video or when the background visual models vary fast in outdoor sequences.

Optical flow based foreground motion detection techniques model the consistency of motion of the foreground objects to distinguish it from the background motion. In [15], Wixson computes the saliency of each pixel based on the straight line distance traveled by it by considering the frame to frame optical flow. The assumption of straight line motion is too restrictive in this case. Mittal et. al. [11] utilizes the optical flow feature to propose a kernel based density estimation technique for background-foreground differentiation. In [3], the consistently moving objects are detected as a group of pixels after compensating for the camera motion. The method in [3] computes consistency in motion and photometry only over immediate frames and may not perform well with short term consistent background motions.

Recently, [10] proposed a novel method for spatio-temporal saliency based on biologically motivated discriminant centre-surround saliency hypothesis. They use the dynamic texture model [6] to represent natural scene dynamics and KL divergence between dynamic texture models of center and surround windows is used to evaluate the saliency of a particular location. Though the algorithm performs well on a variety of complex video sequences, the accuracy of the algorithm is dependent on the size of the center and surround windows. Large foreground objects may not display good discriminant centre-surround saliency in this method. The computational cost of [10] is also large. It is notable that [10] has done a quantitative analysis with other state-of-the-art methods by manually annotating the regions of salient motion over a reasonable number of natural video sequences and evaluating the algorithms on this ground truth. We use the same dataset and ground truth to compare our proposed method with other methods.

Our proposed method for salient object detection also relies on the dynamic texture model for representing natural scenes. The sequence of frames in a video is represented as a Multi Input Multi Output (MIMO) state-space model. We explore the relationships between the output of the model and the states of the model to evaluate the relevancy of every output. The proposed method does not require any kind of explicit background modeling. It is also independent of the size of the object unlike [10]. The major contributions in this paper can be summarized as follows:

1) We introduce the novel concept of evaluating the saliency of the output pixels in a video frame by exploring its relationships to the hidden states when the video sequence is represented as a MIMO state-space model.
2) The state space model concept of 'observability' from the output is shown to provide a good clue to the salient motion in natural videos.
3) The proposed algorithmic framework is tested on a variety of natural video sequences and is quantitatively evaluated against the state-of-the-art methods.

The paper is organized as follows. Section 2 reviews the theory of dynamic textures and the observability of a state space model. Section 3 discusses the

proposed algorithm by first establishing the relationship between the observability of the linear dynamical system and salient motion in a video. The method of computing saliency map by evaluating observability from different outputs is further elaborated in Section 3. Section 4 is the experiments section with subjective results and objective comparisons with other methods. Finally, conclusions are given in Section 5.

2 State-Space Representation

In this section we will review some theory about the state space representation of natural video sequences and the control theory concept of observability of state space models.

2.1 Dynamic Textures

In [6], Doretto et. al proposed a state space model known as dynamic textures, in which the frames in a video sequence are represented as the output of a linear dynamical system. We also represent the frames in the video sequence using this state space model in our proposed method as it can effectively capture the complex spatio-temporal dynamics of many naturally occuring scenes. The output of the model at time t is the frame vector observed at time t, $y(t) \in \mathbb{R}^m$, which is the result of a hidden state markov process driven by a gaussian observation input noise. The auto-regressive model can be represented as

$$\begin{aligned} x(t+1) &= \mathbf{A}x(t) + v(t) \\ y(t) &= \mathbf{C}x(t) + w(t) \end{aligned} \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{n \times n}$ is the state transition matrix that defines how the state vector $x(t) \in \mathbb{R}^n$ evolves. $v(t)$ is the input gaussian observation noise which is an IID realization from the density, $\mathcal{N}(0, \mathbf{Q})$,where $\mathbf{Q} \in \mathbb{R}^{n \times n}$ is the covariance matrix of the zero-mean gaussian process. $w(t)$ is the output observation noise sampled from $\mathcal{N}(0, \mathbf{R})$, where $\mathbf{R} \in \mathbb{R}^{m \times m}$ is the covariance matrix. The $m \times n$ matrix $\mathbf{C}$ relates the current state to the observed output. The system identification problem proposed by [10] estimates the matrices $\mathbf{A}$,$\mathbf{C}$,$\mathbf{Q}$ and $\mathbf{R}$ from the measurements $y(1)$,$y(2)...y(\tau)$. Let $\mathbf{Y}_1^\tau = [y(1), y(2), ..y(\tau)] \in \mathbb{R}^{m \times \tau}$ and $\mathbf{X}_1^\tau = [x(1), x(2), ..x(\tau)] \in \mathbb{R}^{n \times \tau}$ be the matrices stacked with observed output vectors and the hidden state vectors respectively. [10] estimates $\mathbf{C}(\tau)$ and $\mathbf{X}_1^\tau$ as

$$\begin{aligned} \mathbf{C}(\tau) &= U \\ \mathbf{X}_1^\tau &= \Sigma V \end{aligned} \tag{2}$$

where $\mathbf{Y}_1^\tau = U\Sigma V$, is the SVD decomposition of the stacked up output observation vectors. The state transition matrix at time τ, $\mathbf{A}(\tau)$ is determined as the unique solution to the linear problem $\mathbf{A}(\tau) = \texttt{argmin}_A \left\| \mathbf{X}_2^\tau - A\mathbf{X}_1^{\tau-1} \right\|_F$ where $\|.\|_F$ is the Frobenius norm. The variance of state estimates are further used to estimate the input noise covariance matrix $\mathbf{Q}$.

A video sequence model estimated in this manner results in a holistic representation of the motion of the scene. The appearance component of the scene is encoded in to the matrix $\mathbf{Y}_1^\tau$ while the motion component is encoded into the state sequence [4]. Further, the columns of matrix $\mathbf{C}(\tau)$ consists of the principal components of the video sequence in the time period $t = 1$ to τ and the output at time τ, $y(\tau)$, is the linear combination of columns of $\mathbf{C}(\tau)$ weighted by the appropriate state estimates from the vector $x(\tau)$.

2.2 Observability of a State Space Model

Observability of a state space model is a concept from control theory which measures how well the internal states of a system can be inferred with the knowledge of its external outputs. Consider the following more generic form of Equation (1).

$$\begin{aligned} x(t+1) &= \mathbf{A}x(t) + \mathbf{B}v(t) \\ y(t) &= \mathbf{C}x(t) + \mathbf{D}w(t) \end{aligned} \tag{3}$$

Formally, the state-space model in Equation (3) is said to be observable if any unknown initial state $x(0)$ can be uniquely determined by the knowledge of output $y(t)$ along with the knowledge of the input $u(t)$ in a finite time interval $[0\ \ t_1]$. In simpler terms, observability relates the observed outputs of a system to its hidden internal states. If a system is not observable it implies that the current values of some of its internal states cannot be estimated from its current or future outputs.

The observability of a linear time invariant system can be easily determined from the rank conditions on the $nm \times n$ 'observability matrix',O, formed by the $n \times n$ matrix $\mathbf{A}$ and $m \times n$ matrix $\mathbf{C}$ as,

$$O = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ . \\ . \\ \mathbf{CA}^{n-1} \end{bmatrix}. \tag{4}$$

The system described by $(\mathbf{A}, \mathbf{C})$ is completely observable when the $nm \times n$ observability matrix O has full column rank n. It has to be further noted that the basis of a state space representation is arbitrary with any invertible linear transform, $\mathbf{T} \in \mathbb{R}^{n \times n}$, acting on the states of the system as $\mathbf{T}x$, can result in an equivalent representation with a different set of system parameters: $\bar{\mathbf{A}} = \mathbf{TAT}^{-1}, \bar{\mathbf{B}} = \mathbf{TB}, \bar{\mathbf{C}} = \mathbf{CT}^{-1}, \bar{\mathbf{D}} = \mathbf{D}$ [5]. This *equivalence class* of systems will have same transfer function matrix that relates the inputs and outputs. Also the eigen values of $\mathbf{A}$ and $\bar{\mathbf{A}}$ remain the same for any invertible $\mathbf{T}$. Any specific eigen value of state transition matrix $\mathbf{A}$, say λ_i, will be present in equation of output $y(t)$ depending on whether λ_i is observable or not. Observability of a system remains unchanged in all forms of its equivalent representations. Hence, observability of eigen values of $\mathbf{A}$ becomes the final factor that decides the system observability irrespective of a specific realization among all possible realizations

in the equivalence class. Only observable eigen values of matrix A appear on the final output equation of the state space system [2].A state space system is said to be completely observable if all the eigen values of **A** are observable. We can summarize this short discussion on observability by noting down three related aspects of it as i) observability of the system as decided by the rank of the observability matrix O ii) observability of the state variables or eigen values of the state transition matrix **A** iii) observabilty from the outputs of the system to the eigen values of **A**. We are specifically interested in the third factor since the observability from the outputs of a system has strong relation to salient motion as will be seen in the next section.

3 Observability from Outputs and Salient Motion

Now we move on to explore the relationship between the state space concept of observability and salient motion in a video sequence when the sequence is represented in the state space framework described in Section 2.1. Consider the video sequences with various dynamic backgrounds shown in figure 1. The 'jump' sequence in figure 1(a) has smoke filled environment as the background, while the 'ocean' sequence in figure 1(b) has waves in the sea as the dynamic background. The continuous snow fall and the snow thrown out from the skiing gear serves as the complex dynamic background for the 'skiing' video sequence in figure 1(c).

The salient or foreground motion in these three different scenarios can be inferred as the cyclist jumping, the people walking and the skiers moving, respectively. This inference is made in the presence of complex and dynamic

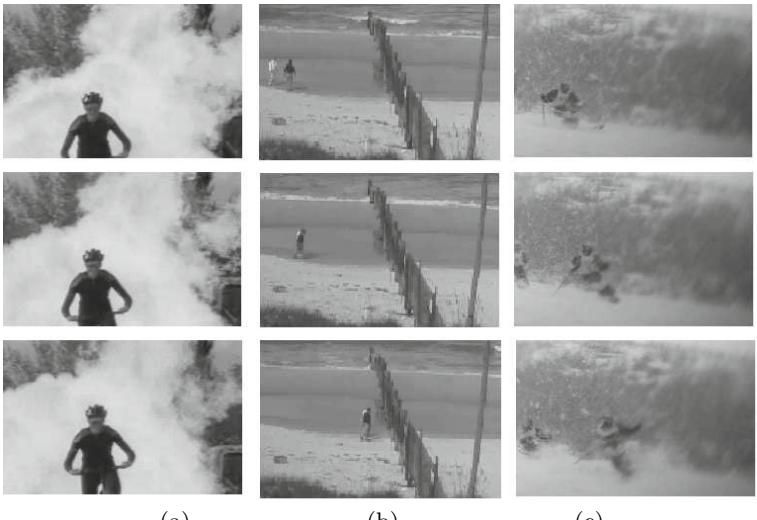


Fig. 1. Video sequences with different dynamic backgrounds (a) smoke (b) waves (c) snow fall

background in all the three cases. The attentive motion is seen as the more 'predictable' or 'easy to observe' one for a human observer compared to the relatively unpredictable backgrounds. For example, the smoke moving in the background of the 'jump' sequence is haphazard in its motion, while the cyclist keeps a regular motion. Similarly, the movement of skiers is more robust and predictable compared to the snow fall and the motion of snow thrown out while skiing. Though waves in figure 1(b) are less complex in its motion, the walking of people becomes a relatively simpler motion compared to waves. So, we attach the notion of 'predictability' or 'regularity' to the salient motion. It is not as restrictive as the notion of linear motion as in [15] and can handle various kinds of motion. Also we only intend to focus on relative predictability as in the case of figure 1(b).

As explained in Section 2.1, the dynamic texture model represents a scene of τ frames in a holistic manner using the state space framework where the matrix $\mathbf{C}$ is the appearance matrix which encodes the visual appearance of the texture and the states $x(t)$ encode the motion of the entire sequence [4]. The observability from a specific pixel in output $y(t)$ is a measure of inference that can be made on the values of state variables which is a holistic characterization of the motion involved in the video sequence. Pixels belonging to complex unpredictable backgrounds will have less observability to the state variables compared to the pixels belonging to regions of simple predictable motions. This is due to the fact that we are modeling the video sequence over a period of time (τ frames), and the motion components belonging to complex unpredictable regions are hard to estimate from its holistic characterization. Pixels belonging to static regions will have zero observability as they do not contribute to the inference of any motion and hence the inference of state variables. Hence observability from a pixel computed under the framework in Equation (1) can be concluded as a good clue for saliency of motion.

3.1 Quantitative Measure for Observability

In Section 2.2 we reviewed the basic definition of observability and a simple rank based method from control theory literature to evaluate whether a system is observable or not. However such methods only help to make a binary decision on the observability of a system and do not provide any insight into the observability from different outputs to the eigenvalues of state transition matrix $\mathbf{A}$ (Note that observability is evaluated with respect to eigenvalues of $\mathbf{A}$ as it is invariant under various equivalent system representations). In [14], Tarokh discusses a simple and computationally efficient method to evaluate m_{oi}, a measure of observability for the i^{th} eigen value λ_i of the state transition matrix $\mathbf{A}$. It is calculated as [14],

$$m_{oi} = |\delta_i| \, [e_i^* \mathbf{C}^T \mathbf{C} e_i]^{\frac{1}{2}}. \tag{5}$$

where e_i is the right eigen vector of the transition matrix $\mathbf{A}$ corresponding to λ_i and δ_i is calculated as

$$|\delta_i| = | \prod_{j=1, i \neq j}^{n} (\lambda_i - \lambda_j)| \tag{6}$$

(a) (b) (c) (d) (e)

Fig. 2. Example showing the generation of final saliency map with sustained observability a) 12^{th} frame of 'cyclists' video (b, c, d) saliency maps s1, s2 and s3 computed using frame buffers with frames 1 to 21, 2 to 22 and 3 to 23 respectively e) final saliency map

It can be further concluded from Equation (5) that the angle between a specific row of output matrix $\mathbf{C}$ and the right eigen vector e_i is a measure of observability from the respective output. If all the row vectors of the output matrix $\mathbf{C}$ are orthogonal to the right eigen vector e_i, that specific λ_i will have zero observability [14]. We are particularly interested in the measure m_{pi} which indicates how much useful is output p in inferring on eigen value λ_i. In our proposed method every output will correspond to a pixel p in the output frame as $y(t)$ is the output frame reshaped as the $m \times 1$ vector. Hence we evaluate the observability measure from a pixel p to λ_i as

$$m_{pi} = [\mathbf{C}e_i]_p \tag{7}$$

where $[\mathbf{C}e_i]_p$ is the scalar component of the vector $\mathbf{C}e_i$ corresponding to output p. The total observability from output p, denoted by m_p ,is calculated as the sum of its observability measures to all the eigen values of $\mathbf{A}$ i.e.,

$$m_p = \sum_{i=1}^{n} m_{pi} \tag{8}$$

We evaluate the observability from all the pixels in the output in a similar fashion and normalize the values between 0 and 1 to get the saliency map.

3.2 Sustained Observability

The observability measure evaluated for the pixels in the output vector indeed provides a good clue for salient motion. However in this process some other pixels in the dynamic background can also get some observability as can be seen in the 'cyclists' video sequence whose 12^{th} frame is shown in figure 2(a). Figure 2(b), (c) and (d) show the saliency maps for three adjacent frame buffers wherein some part of the dynamic backgound also acquires observability. The concept of sustained observability is to get the points of consistent performance irrespective of the potential errors that would have happened due to the sub-optimal system modeling using [6] for a single buffer. We look for pixels having 'sustained observability', i.e. those pixels that maintain high observability over succesive saliency maps computed for adjacent frame buffers as shown in figure 3.

We evaluate the dynamic texture models for adjacent buffers using frames 1 to q, 2 to q+1 and 3 to q+2 respectively. s1, s2, s3 are the saliency maps computed

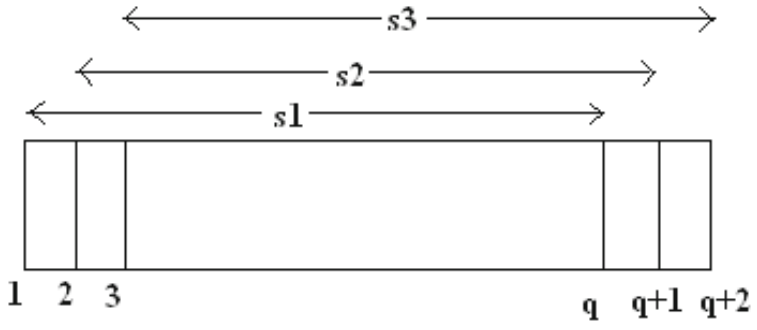


Fig. 3. Sustained observability calculated over adjacent video buffers. s1 , s2 and s3 are saliency maps computed using dynamic texture models from frames 1 to q, 2 to q+1 and 3 to q+2 respectively.

using the three adjacent models. The final saliency map is computed by multiplying all the three maps and normalizing it in the range [0 1], which implies we give more saliency to those pixels which hold the observability across the three adjacent models. The final saliency map is attributed to the centre input frame in the buffer. It can be verified that though different regions in the dynamic background show some saliency in individual binary saliency maps s1, s2 and s3, the final saliency map in figure 2(e) only shows points that show consistently high observability. Three adjacent buffers are used as an empirical precision-recall trade-off. More adjacent buffers improve precision but affect recall.

4 Experiments

The proposed method is tested on the same experimental database used by [10] available in [20]. The video database used by [10] contains 18 video sequences in varied shooting conditions ranging from surveillance videos with static backgrounds to videos having complex dynamic backgrounds and significant camera motion. [10] has already established a quantitative evaluation method on the video database set using the manually annotated ground truth data and consequently has compared their method with other state-of-the-art methods. Hence by utilizing the same database, ground truth and evaluation methodology we compare and contrast the accuracy of our method with the competing algorithms.

The experiments are done with a buffer size of 21 frames ($q = 21$) and the number of states is empirically fixed as 10($n = 10$). Figure 4 shows three example videos from the dataset and the saliency maps obtained using the proposed sustained observability (**SO**) method. The 'ocean' sequence in figure 4(a) is under static camera condition with the waves as dynamic background while the 'skiing' sequence in figure 4(c) has snow fall background with some camera motion. The 'peds' sequence in figure 4(e) is a typical surveillance video watching pedestrians on a road. The saliency maps in figure 4(b), (d) and (f) show that the proposed

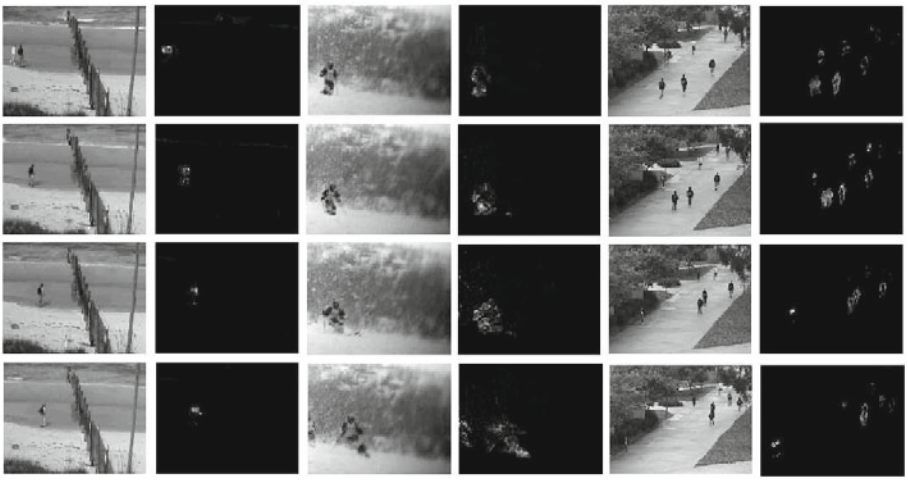


Fig. 4. Saliency maps obtained using the proposed sustained observability method on various video sequences a)'ocean' sequence (waves as background) b) saliency map for 'ocean' sequence c) 'skiing' sequence (snowfall as background) d) saliency map for 'skiing' sequence e) 'peds' sequence (static camera surveillance) f) saliency map for 'peds' sequence

method detects the regions of salient motion accurately under different shooting conditions. Note that no thresholds are applied to the saliency maps shown in figure 4 which indicate actual saliency of pixels obtained using the proposed method.

4.1 Quantitative Evaluation

The objective comparison of different methods is done with the manually annotated ground truth data from [10]. The saliency maps obtained are thresholded at a large number of values for computing the false alarm rate(α) and detection rate(β). The final performance of the algorithm is evaluated by the Equal Error rate (EER), which is the error at which the false alarm rate is equal to the miss rate (α = 1- β). Lesser the EER, better the performance of the algorithm.

We have reproduced the results of objective performance of various methods from [10] for comparison with our proposed method. The proposed sustained observability (SO) method is objectively compared with five different salient motion detection methods - (1) Discriminant Saliency Method (DS) [10] (2)'Surprise' model from Itti and Baldi (Su) [9] (3) Non-parametric Kernel Density Estimator (KDE) method [7] (4) Linear Dynamical Model of Monnet et. al. (Mo) [1] (5) modified Gaussian Mixture Model (GMM) method [17]. Table 1 shows the equal error rates (in %) for saliency maps obtained with different methods for various video sequences available in the database. Table 2 shows the average equal error rates of all competing methods.

Table 1. Equal Error Rates (in %) for different algorithms [10] compared with Sustained Observability (**SO**) method

	DS	Su	KDE	Mo	GMM	**SO**		DS	Su	KDE	Mo	GMM	**SO**
skiing	3	26	46	11	36	**4.9**	bottle	2	5	38	17	25	**3.9**
surf	4	30	36	10	23	**7.6**	hockey	24	28	35	29	39	**27**
cyclists	8	41	44	28	36	**17.9**	land	3	31	54	16	40	**27**
birds	5	19	20	7	23	**9.2**	zodiac	1	19	20	3	40	**5.8**
chopper	5	13	43	8	35	**8.3**	peds	7	37	17	11	11	**7**
flock	15	23	33	31	34	**25**	traffic	3	46	39	9	34	**20**
boat	9	9	13	15	15	**11.3**	freeway	6	43	21	31	25	**8.7**
jump	15	25	51	23	39	**22**	ocean	11	42	19	11	30	**10.8**
surfers	7	24	25	10	35	**9**	rain	3	10	23	17	15	**2**

Table 2. Average Equal Error Rate comparison of proposed method with other state-of-the-art methods

	DS	Su	KDE	Mo	GMM	**SO**
Avg EER	7.6 %	26.2%	33.1 %	16 %	29.7 %	**12.6 %**

4.2 Comparison

It can be seen from Table 2 that the proposed method achieves second best average EER among all the methods compared. Compared to the discriminant saliency (DS) method which has the minimum average EER, the proposed method has significant computational advantage. [10] reports the computational complexity of the DS method as 37 seconds for processing a video frame of size 240×320 pixels with a Matlab implementation on a PC with 3 Ghz CPU, 2 GB RAM. The proposed sustained observability methods takes only approximately 0.25 seconds for processing the video frame of same size with the Matlab implementation on a 2.66 GHz dual core CPU with 2 GB RAM. Hence the proposed method has a large computational advantage over [10] and considerable performance advantage over other competing methods with respect to the average percentage of Equal Error Rates. One shortfall for the proposed method is its increased false alarm rate when the camera follows a foreground object slowly and smoothly with no background motion other than the one induced by the camera. For example, in the 'land' sequence in the video database, the camera follows the landing of an aircraft while the buildings in the static background move slowly in the opposite direction. For the proposed method such a motion in the background is a slow and predictable regular motion and hence will get saliency along with the aircraft in foreground. However the DS method uses contrast of centre and surround windows and gets the motion of aircraft as salient. Hence the 'land' sequence has high EER for the proposed method and low EER for the DS method.

5 Conclusion

We have presented a novel way of defining the salient motion under complex unpredictable backgrounds using the concept of observability from the control theory literature. The proposed method is simple and efficient in a way that it requires no background estimation or understanding of motion in the system. The method is computationally efficient and is shown to provide one of the best results when tested on a dataset with videos having a wide range of motions under different shooting conditions.

Acknowledgement. This research was partially supported by the Media Development Authority under grant NRF2008IDM-IDM004-032.

References

1. Antoine, M., Anurag, M., Nikos, P., Visvanathan, R.: Background Modeling and Subtraction of Dynamic Scenes. In: ICCV, pp. 1305–1312 (2003)
2. Antsaklis, P.J., Michel, A.N.: Linear Systems. McGraw-Hill Higher Education, New York (1997)
3. Bugeau, A., Perez, P.: Detection and segmentation of moving objects in complex scenes. Computer Vision and Image Understanding 113(4), 459–476 (2009)
4. Chan, A.B., Vasconcelos, N.: Probabilistic Kernels for the Classification of Auto-Regressive Visual Processes. In: CVPR, pp. 846–851 (2005)
5. Chen, C.T.: Linear System Theory and Design. Oxford University Press, Inc., NY (1998)
6. Doretto, G., Chiuso, A., Wu, Y.N., Soatto, S.: Dynamic Textures. Int. J. on Computer Vision 51(2), 91–109 (2003)
7. Elgammal, A.M., Duraiswami, R., Harwood, D., Davis, L.S.: Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. Proc. IEEE 90, 1151–1163 (2002)
8. Elgammal, A.M., Harwood, D., Davis, L.S.: Non-parametric model for background subtraction. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1843, pp. 751–767. Springer, Heidelberg (2000)
9. Itti, L., Baldi, P.: A Principled Approach to Detecting Surprising Events in Video. In: CVPR, pp. 631–637 (2005)
10. Mahadevan, V., Vasconcelos, N.: Spatiotemporal Saliency in Dynamic Scenes. IEEE Trans. on Pattern Analysis and Machine Intelligence 32(1), 171–177 (2010)
11. Mittal, A., Paragios, N.: Motion-based background subtraction using adaptive kernel density estimation. In: CVPR, pp. 302–309 (2004)
12. Rubinstein, M., Shamir, A., Avidan, S.: Improved seam carving for video retargeting. ACM Trans. Graph 27(3), 1–9 (2008)
13. Stauffer, C., Grimson, W.: Adaptive Background Mixture Models for Real-Time Tracking. In: CVPR, pp. 2246–2252 (1999)
14. Tarokh, M.: Measures for controllability, observability and fixed modes. IEEE Transactions on Automatic Control 37(8), 1268–1273 (1992)
15. Wixson, L.E.: Detecting Salient Motion by Accumulating Directionally-Consistent Flow. IEEE Trans. on Pattern Analysis and Machine Intelligence 22(8), 774–780 (2000)

16. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Comput. Surv. 38(4) (2006)
17. Zivkovic, Z.: Improved adaptive gaussian mixture model for background subtraction. In: ICPR, vol. II, pp. 28–31 (2004)
18. Zhong, J., Sclaroff, S.: Segmenting Foreground Objects from a Dynamic Textured Background via a Robust Kalman Filter. In: ICCV, pp. 44–50 (2003)
19. Zhu, J., Lao, Y., Zheng, Y.F.: Object Tracking in Structured Environments for Video Surveillance Applications. IEEE Transactions on Circuits and Systems for Video Technology 20(2), 223–235 (2010)
20. http://www.svcl.ucsd.edu/projects/background_subtraction

Markerless and Efficient 26-DOF Hand Pose Recovery

Iasonas Oikonomidis, Nikolaos Kyriazis, and Antonis A. Argyros

Institute of Computer Science, FORTH
and
Computer Science Department, University of Crete
{oikonom,kyriazis,argyros}@ics.forth.gr
http://www.ics.forth.gr/cvrl/

Abstract. We present a novel method that, given a sequence of synchronized views of a human hand, recovers its 3D position, orientation and full articulation parameters. The adopted hand model is based on properly selected and assembled 3D geometric primitives. Hypothesized configurations/poses of the hand model are projected to different camera views and image features such as edge maps and hand silhouettes are computed. An objective function is then used to quantify the discrepancy between the predicted and the actual, observed features. The recovery of the 3D hand pose amounts to estimating the parameters that minimize this objective function which is performed using Particle Swarm Optimization. All the basic components of the method (feature extraction, objective function evaluation, optimization process) are inherently parallel. Thus, a GPU-based implementation achieves a speedup of two orders of magnitude over the case of CPU processing. Extensive experimental results demonstrate qualitatively and quantitatively that accurate 3D pose recovery of a hand can be achieved robustly at a rate that greatly outperforms the current state of the art.

1 Introduction

The problem of effectively recovering the pose (3D position and orientation) of human body parts observed by one or more cameras is interesting because of its theoretical importance and its diverse applications. The human visual system exhibits a remarkable ability to infer the 3D body configurations of other humans. A wide range of applications such as human-computer interfaces, etc, can be built provided that this fundamental problem is robustly and efficiently solved [1]. Impressive motion capture systems that employ visual markers [2] or other specialized hardware have been developed. However, there is intense interest in developing markereless computer-vision based solutions, because they are non-invasive and, hopefully, cheaper than solutions based on other technologies (e.g., electromagnetic tracking).

The particular problem of 3D hand pose estimation is of special interest because by understanding the configuration of human hands we are in a position

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 744–757, 2011.

to build systems that may interpret human activities and understand important aspects of the interaction of a human with her/his physical and social environment. Despite the significant amount of work in the field, the problem remains open and presents several theoretical and practical challenges due to a number of cascading issues. Fundamentally, the kinematics of the human hand is complicated. Complicated kinematics is hard to accurately represent and recover and also yields a search space of high dimensionality. Extended self-occlusions further complicate the problem by generating incomplete and/or ambiguous observations.

1.1 Related Work

A significant amount of literature has been devoted to the problem of pose recovery of articulated objects using visual input. Moeslund et al [1] provide a thorough review covering the general problem of visual human motion capture and analysis. The problems of recovering the pose of the human body and the human hand present similarities such as the tree-like connectivity and the size variability of the articulated parts. However, a human hand usually has consistent appearance statistics (skin color), whereas the appearance of humans is much more diverse because of clothing.

A variety of methods have been proposed to capture human hand motion. Erol et al [3] present a review of such methods. Based on the completeness of the output, they differentiate between partial and full pose estimation methods, further dividing the last class into appearance-based and model-based ones.

Appearance-based methods estimate hand configurations from images directly after having learnt the mapping from the image feature space to the hand configuration space [4,5,6,7]. The mapping is highly nonlinear due to the variation of hand appearances under different views. Further difficulties are posed by the requirement for collecting large training data sets and the accuracy of pose estimation. On the positive side, appearance based methods are usually fast, require only a single camera and have been successfully employed for gesture recognition.

Model-based approaches employ a 2D or 3D hand model [8,9,10,11]. In the case of 3D hand models the hand pose is estimated by matching the projection of the model to the observed image features. The task is then formulated as a search problem in a high dimensional configuration space, which induces a high computational cost. Important issues to be addressed by such methods include the efficient construction of realistic 3D hand models, the dimensionality reduction of the configuration space and the development of techniques for fast and reliable hand posture estimation.

This paper presents a novel, generative method that treats the 3D hand pose recovery problem as an optimization problem that is solved through Particle Swarm Optimization (PSO). Under the taxonomy of [3], the present work can be categorized as a full, model-based pose estimation method that employs a single hypothesis. The method may integrate observations from an arbitrary number of available views without requiring special markers. This is clearly demonstrated by our decision to consider all free problem parameters jointly and simultaneously.

As a direct consequence, contrary to the work of [10], our formulation of the problem allows for a clear and effortless treatment of self-occlusions. PSO has been already applied for human pose recovery in [12], however this is done in a hierarchical fashion in contrast to our joint optimization approach. Additionally, the method of [12] is not directly applicable to hand pose recovery because stronger occlusions must be handled given weaker observation cues.

Being generative, the approach explores an essentially infinite configuration space. Thus, the accuracy of estimated pose is not limited by the size and content of the employed database, as e.g. in [7]. To the best of our knowledge, this is the first work that demonstrates that PSO can be applied to the problem of 3D hand pose recovery and solve it accurately and robustly. This is demonstrated in sequences with highly complex hand articulation where the hand is observed from relatively distant views. Additionally, it is demonstrated that the careful selection of inherently data parallel method components permits the efficient, near real-time 3D hand pose estimation and gives rise to the fastest existing method for model-based hand pose recovery.

The rest of this paper is organized as follows. Section 2 describes in detail the proposed method. Section 3 presents results from an extensive quantitative and qualitative experimental evaluation of the proposed method. Finally, Sec. 4 summarizes the paper by drawing the most important conclusions of this work.

2 Methodology

The proposed method can be summarized as follows. Observations of a human hand are acquired from a static, pre-calibrated camera network. For each observation, skin color detection and edge detection are performed to extract reference features. A 3D model of a human hand is adopted that consists of a collection of parameterized geometric primitives. Hand poses are represented by a total of 27 parameters that redundantly encode the 26 degrees of freedom of the human hand. Given the hand model, poses which would reproduce the observations are hypothesized. For each of them, the corresponding skin and edge feature maps are generated and compared against their reference counterparts. The discrepancy between a given pose and the actual observation is quantified by an error function which is minimized through Particle Swarm Optimization (PSO). The pose for which this error function is minimal constitutes the output of the proposed method at a given moment in time. Temporal continuity in hand motion is assumed. Thus, the initial hypotheses for current time instance are restricted in the vicinity of the solution for the previous time instant. The method incorporates computationally expensive processes which cannot be adequately handled by conventional CPU processing. However, the exploitation of the inherent data parallelism of all the required components through a GPU powered implementation, results in near real-time computational performance. The following sections describe in more detail the components outlined above.

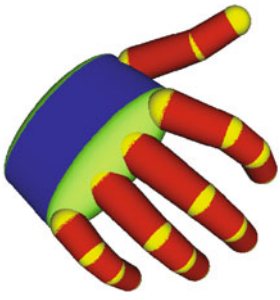


Fig. 1. Hand model with colored parts. Each color denotes a different type of geometric primitive (blue for elliptic cylinders, green for ellipsoids, yellow for spheres and red for cones).

2.1 Observation Model

The proposed hand pose recovery method operates on sequences of synchronized views acquired by intrinsically and extrinsically calibrated cameras. A set of images acquired from a set of such cameras at the same moment in time is called a *multiframe*. If $M_i = \{I_1, I_2, \ldots\}$ is a multiframe of a sequence $S = \{M_1, M_2, \ldots\}$ then I_j denotes the image from the j-th camera/view at the i-th time step. In the single camera case, a sequence of multiframes reduces to an image sequence.

An observation model similar to [10] is employed. For each image I of a multiframe M, an edge map $o_e(I)$ is computed by means of Canny edge detection [13] and a skin color map $o_s(I)$ is computed using the skin color detection method employed in [14]. As a convention, the label of 1 indicates presence and the label of 0 indicates the absence of skin or edges in the corresponding maps. For each edge map $o_e(I)$, a distance transform $o_d(I)$ is computed. For each image I, maps $O(I) = \{o_s(I), o_d(I)\}$ constitute its observation cues.

2.2 Hand Model

The model of hand kinematics used in this work is based on [15]. The kinematics of each finger, including the thumb, is modeled using four parameters encoding angles. More specifically, two are used for the base of the finger and two for the remaining joints. Bounds on the values of each parameter are set based on anatomical studies (see [15] and references therein). The global position of the hand is represented using a fixed point on the palm. The global orientation is parameterized using the redundant representation of quaternions. This parameterization results in a 26-DOF model encoded in a vector of 27 parameters.

The hand consists of a palm and five fingers. The palm is modeled as an ellipsoid cylinder and two ellipsoids for caps. Each finger consists of three cones and four spheres, except for the thumb which consists of two cones and three spheres (see Fig. 1). All required 3D shapes used in the adopted hand model consist of multiple instances of two basic geometric primitives, a sphere and a truncated cylinder. These geometric primitives, subjected to appropriate

homogeneous transformations, yield a model similar to that of [9]. Each transformation performs two different tasks. First, it appropriately transforms primitives to more general quadrics and, second, it applies the required kinematics. Using the shape transformation matrix

$$T_s = \begin{pmatrix} e \cdot s_x & 0 & 0 & 0 \\ 0 & e \cdot s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 1-e & e \end{pmatrix}, \tag{1}$$

spheres can be transformed to ellipsoids and cylinders to elliptic cylinders or cones. In Eq.(1), s_x, s_y and s_z are scaling factors along the respective axes. The parameter e is used only in the case of cones, representing the ratio of the small to the large radius of the cone before scaling (if not transforming to a cone, e is fixed to 1). Having a rigid transformation matrix T_k computed from the kinematics model, the final homogeneous transformation T for each primitive (sphere or cylinder) is

$$T = T_k \cdot T_s. \tag{2}$$

A non-trivial implementation issue (see Sec. 2.5) is the correct computation of surface normals. For given normals $\overrightarrow{n_i}$ of the two primitives in use, and given homogeneous transformation T, the computation of the new surface normals $\overrightarrow{n_i}'$ can be performed according to [16] using the equation $\overrightarrow{n_i}' = (T^{-\mathrm{T}})_{3\times3} \cdot \overrightarrow{n_i}$. $A_{3\times3}$ denotes the upper-left 3 by 3 submatrix of A.

Having a parametric 3D model of a hand, the goal is to estimate the model parameters that are most compatible to the observed images/image features (Sec. 2.1). To do so, we compute comparable image features from each hypothesized 3D hand pose (see Sec. 2.5). More specifically, given a hand pose hypothesis h, an edge map $r_e(h)$ and a skin color map $r_s(h)$ can be generated my means of rendering. The reference implementation of the rendering process is very similar to that of [9]. The informative comparison between each observation and corresponding hypotheses is detailed in Sec. 2.3.

2.3 Hypothesis Evaluation

The proposed method is based on a measure quantifying how compatible a given 3D hand pose is to the actual camera-based observations. More specifically, a distance measure between a hand pose hypothesis h and the observations of multiframe M needs to be established. This is performed by the computation of a function $E(h, M)$ which measures the discrepancies between skin and edge maps computed in a multiframe and the skin and edge maps that are rendered for a given hand pose hypothesis:

$$E(h, M) = \sum_{I \in M} D(I, h, \mathrm{C}(I)) + \lambda_k \cdot kc(h). \tag{3}$$

In Eq.(3), h is the hand pose hypothesis, M is the corresponding observation multiframe, I is an image in M, $\mathrm{C}(I)$ is the set of camera calibration parameters

corresponding to image I and λ_k is a normalization factor. The function D of Eq.(3) is defined as

$$D(I,h,c) = \frac{\sum o_s(I) \otimes r_s(h,c)}{\sum o_s(I) + \sum r_s(h,c) + \epsilon} + \lambda \frac{\sum o_d(I) \cdot r_s(h,c)}{\sum r_e(h,c) + \epsilon}, \tag{4}$$

where $o_s(I), o_d(I), r_s(h,c), r_e(h,c)$ are defined in Sec. 2.1. A small term ϵ is added to the denominators of Eq.4) to avoid divisions by zero. The symbol $\otimes$ denotes the logical XOR (exclusive disjunction) operator. Finally, λ is a constant normalization factor. The sums are computed over entire feature maps. The function kc adds a penalty to kinematically implausible hand configurations. Currently, only adjacent finger inter-penetration is penalized. Therefore, kc is defined as

$$kc(h) = \sum_{p \in \text{pairs}} \begin{cases} -\phi(p) & \phi(p) < 0 \\ 0 & \phi(p) \geq 0 \end{cases}, \tag{5}$$

where *pairs* denotes the three pairs of adjacent fingers, excluding the thumb, and ϕ denotes the difference between the abduction-adduction angles of those fingers. In all experiments the values of λ and λ_k were both set to 10.

2.4 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is an optimization technique that was introduced by Kennedy et al [17]. It is an evolutionary algorithm since it incorporates concepts such as populations, generations and rules of evolution for the atoms of the population (particles). A population is essentially a set of particles which lie in the parameter space of the objective function to be optimized. The particles evolve in runs which are called generations according to a policy which emulates "social interaction".

Canonical PSO, the simplest of PSO variants, was preferred among other optimization techniques due to its simplicity and efficiency. More specifically, it only depends on very few parameters, does not require extra information on the objective function (e.g., its derivatives) and requires a relatively low number of evaluations of the objective function [18]. Following the notation introduced in [19], every particle holds its current position (current candidate solution, set of parameters) in a vector x_t and its current velocity in a vector v_t. Moreover, each particle i stores in vector p_i the position at which it achieved, up to the current generation t, the best value of the objective function. Finally, the swarm as a whole, stores in vector p_g the best position encountered across all particles of the swarm. p_g is broadcasted to the entire swarm, so that every particle is aware of the global optimum. The update equations that are applied in every generation t to reestimate each particle's velocity and position are

$$v_t = K(v_{t-1} + c_1 r_1 (p_i - x_{t-1}) + c_2 r_2 (p_g - x_{t-1})) \tag{6}$$

and

$$x_t = x_{t-1} + v_t, \tag{7}$$

where K is a constant *constriction factor* [20]. In Eqs. (6), c_1 is called the *cognitive component*, c_2 is termed the *social component* and r_1, r_2 are random samples of a uniform distribution in the range [0..1]. Finally, $c_1 + c_2 > 4$ must hold [20]. In all performed experiments the values $c_1 = 2.8$, $c_2 = 1.3$ and $K = \frac{2}{\left|2-\psi-\sqrt{\psi^2-4\psi}\right|}$ with $\psi = c_1 + c_2$ were used.

Typically, the particles are initialized at random positions and their velocities to zero. Each dimension of the multidimensional parameter space is bounded in some range. If, during the position update, a velocity component forces the particle to move to a point outside the bounded search space, this component is zeroed and the particle does not perform any move at the corresponding dimension. This is the only constraint employed on velocities.

In this work, the search space is the 27-dimensional 3D hand pose parameter space, the objective function to be minimized is $E(M, h)$ (see Eq.(3)) and the population is a set of candidate 3D hand poses hypothesized for a single multiframe. Thus the process of tracking a hand pose requires the solution of a sequence of optimization problems, one for each of the acquired multiframes. By exploiting temporal continuity, the solution over multiframe M_t is used to generate the initial population for the optimization problem of M_{t+1}. More specifically, the first member of the population h_{ref} for M_{t+1} is the solution for M_t; The rest of the population consists of perturbations of h_{ref}. Since the variance of these perturbations depends on the image acquisition frame rate and the anticipated jerkiness of the observed hand motion, it has been experimentally determined in the reported experiments. The optimization for multiframe M_{t+1} is executed for a fixed amount of generations/iterations. After all generations have evolved, the best hypotheses h_{best} is dubbed as the solution for time step $t + 1$.

2.5 Exploiting Parallelism

A reference implementation of the proposed method was developed in MATLAB. A study of the computational requirements of the method components revealed that PSO and skin color detection are very fast. The computations of edge maps and their distance transforms are relatively slow but these tasks along with skin color detection are only executed once per multiframe. The identified computational bottlenecks are the rendering of a given 3D hand pose hypothesis and the subsequent evaluation of Eq.(3). More specifically, the hand model consists of a series of quadrics for which ray casting is used for rendering [9]. Additionally, since multiple quadrics overlap on the projection plane, pixel overwriting will occur and z-buffering is required so as to produce correct edge maps. The computation of Eq.(3) is a matter of pixel-wise multiplication and summation over entire images. The whole process is computationally expensive and prevents real-time performance. Reasonable PSO parameterizations where particles and generations range in the orders of tens, correspond to more than 4 minutes of processing time per multiframe.

GPU accelerated observation models have been employed in the past (e.g. [21]). In contrast to previous work, we provide a detailed description of a GPU implementation that exploits parallelism beyond the point of straightforward

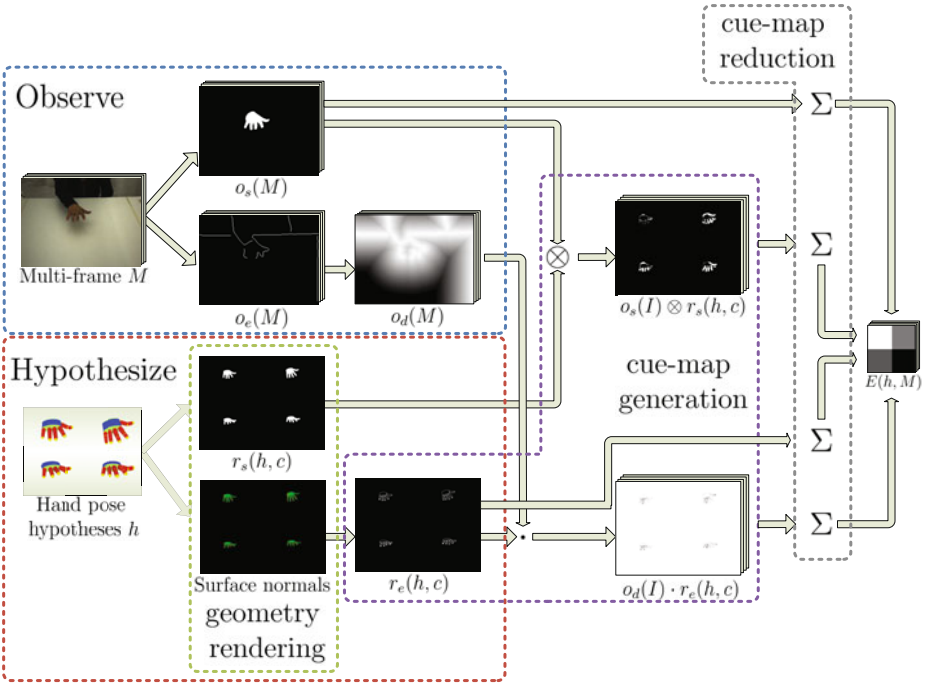


Fig. 2. Back-projection error computation flowchart. Observations of a human hand and hypothesized 3D poses are compared. Reference features are extracted from multiframes by means of skin color detection and edge detection. Artificial features are generated for the 3D pose hypotheses by means of rendering and edge detection. The three main GPU steps are annotated: geometry rendering, cue-map generation and cue-map reduction.

image processing and rendering. Our GPU implementation targets the acceleration of the two performance bottlenecks, i.e., rendering and evaluation. The rest of the tasks are also susceptible to acceleration (e.g. [22,23,24]) but this was not considered in this work. The final implementation used the Direct3D rendering pipeline to accelerate the computationally demanding tasks and MATLAB to perform the rest of the tasks as well as overall task coordination.

Rendering and evaluation of Eq.(3) are decomposed in three major GPU computation steps: geometry rendering, cue-map generation and cue-map reduction (see Fig. 2). Multiple particles are evaluated in large batches instead of single particles. This design choice defines a fine parallelization granularity which makes GPUs the optimal accelerator candidate.

Geometry rendering. The goal of the geometry rendering step is to simultaneously render multiple hand hypotheses in a big tiled rendering. Multiple renderings, instead of sequences of single renderings, were preferred in order to maximally occupy the GPU cores with computational tasks. The non-trivial issues to address are geometry instancing and multi-viewport clipping.

Hardware instancing [24] is used to perform multiple render batches efficiently. Efficiency regards both optimal GPU power exploitation and minimal memory usage. Batch rendering of multiple hand configurations essentially amounts to rendering of multiple instances of spheres and cylinders. However, the respective geometric instantiations are not required to be explicit. Hardware geometry instancing can be used in order to virtually replicate reusable geometry and thus make instantiation implicit.

A specialized pixel shader is used in order to perform custom multi-viewport clipping. Multiple viewports are required to be simultaneously rendered. However, conventional rendering pipelines do not account for multiple viewports, except for the case of sequential renderings. Unless multi-viewport clipping was performed, out of bounds geometry would expand beyond the tiles and spoil adjacent renderings.

The information that is transferred from CPU to GPU are the projection matrices c for each tile and the view matrix T for each primitive. The output of this rendering is the map $r_s(h, c)$, per pixel depth and per pixel normal vectors, encoded in four floating point numbers.

Cue-map generation. During cue-map generation, the output of the geometry rendering step is post-processed in order to provide cue-maps $r_s(h, c)$, $r_e(h, c)$, $o_s(I) \otimes r_s(h, c)$ and $o_e(I) \cdot r_e(h, c)$ of Eq.(3). Cue-map $r_s(h, c)$ passes through this stage since it is computed during geometry rendering (see Fig. 2). Cue-map $r_e(h, c)$ is computed by thresholding the discontinuity in normal vectors for a cross-neighborhood around each pixel. Cue-maps $o_s(I) \otimes r_s(h, c)$ and $o_e(I) \cdot r_e(h, c)$ are trivially computed by element wise operations between the operands.

Cue-map reduction. In the cue-map reduction step, scale space pyramids are employed to efficiently accumulate values across tiles. The expected input is an image that encodes maps $r_s(h, c)$, $r_e(h, c)$, $o_s(I) \otimes r_s(h, c)$ and $o_e(I) \cdot r_e(h, c)$ and the expected output is the sum over logical tiles of these maps. The pyramids are computed by means of sub-sampling, which is a very efficient GPU computation. Once the sums have been accumulated, the computation of Eq.(3) is straightforward.

3 Experimental Evaluation

The quantitative and qualitative experimental validation of the proposed method was performed based on both synthetic and real-world sequences of multiframes.

3.1 Quantitative Evaluation Based on Synthetic Data

The quantitative evaluation of the proposed method was based on synthetic sequences of multiframes which make possible the assessment of the proposed method against known ground truth. Towards this end, the hand model presented in Sec. 2.2 was animated so as to perform motions as simple as waving

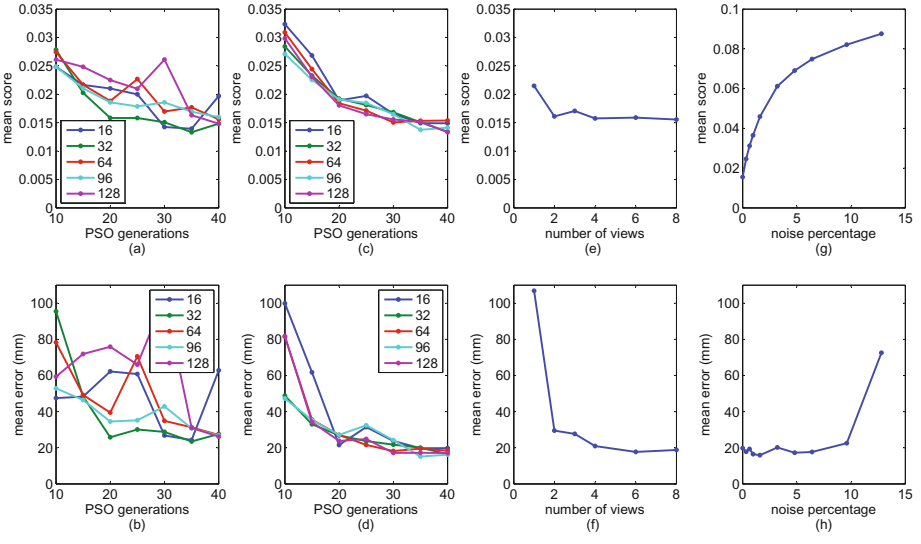


Fig. 3. Performance of the proposed method for different values of selected parameters. In the plots of the top row, the vertical axis represents the mean score E. In the plots of the bottom row, the vertical axis represents mean error in mm (see text for additional details). (a),(b): Varying values of PSO particles and generations for 2 views. (c),(d): Same as (a),(b) but for 8 views. (e),(f): Increasing number of views. (g),(h): Increasing amounts of noise.

and as complex as object grasping. A synthetic sequence of 360 poses of the moving hand was created. Each pose was observed by 8 virtual cameras surrounding the hand. This results in a sequence of 360 multiframes of 8 views, which constitute the input to the proposed method. The required cue maps were synthesized through rendering (see Sec. 2.2).

The performed quantitative evaluation assessed the influence of several factors such as PSO parameters, number of available views (i.e., multiframe size) and segmentation noise, over the performance of the proposed method. Figure 3 illustrates the obtained results. For each multiframe of the sequence, the best scoring hand pose h_{best} using the specified parameter values was found. Figures 3(a), (c), (e) and (g) provide plots of the score $E(h_{best}, M)$ (averaged for all multiframes M) as a function of various experimental conditions. Similarly, Figs. 3(b), (d), (f) and (h) illustrate the actual error in 3D hand pose recovery in millimeters, in the experimental conditions of Figs. 3(a), (c), (e) and (g), respectively. This error was computed as follows. The five fingertips as well as the center of the palm were selected as reference points. For each such reference point, the Euclidean distance between its estimated position and its ground truth position was first calculated. These distances were averaged across all multiframes, resulting in a single error value for the whole sequence.

Figures 3(a) and (b) show the behavior of the proposed method as a function of the number of PSO generations and particles per generation. In this

experiment, each multiframe consisted of 2 views with no noise contamination. It can be verified that varying the number of particles per generation does not affect considerably the error in 3D hand pose recovery. Thus, the number of generations appears to be more important than the number of particles per generation. Additionally, it can be verified that the accuracy gain for PSO parameterizations with more than 16 particles and more than 25 generations was insignificant. Figures 3(c), (d) are analogous to those of Figs 3(a),(b), except the fact that each multiframe consisted of 8 (rather than 2) views. The error variance is even smaller in this case as a consequence of the increased number of views which provides richer observations and, thus, more constraints. The accuracy gain for PSO parameterizations with more than 16 particles and more than 25 generations is even less significant.

In order to assess the behavior of the method with respect to the number of available views of the scene, experiments with varying number of views were conducted. Figures 3(e) and (f) show the behavior of the proposed method as a function of the size of a multiframe. For the experiments with less than 8 views, these were selected empirically so as to be as complementary as possible. More specifically, views with large baselines and viewing directions close to vertical were preferred. In these experiments, 128 PSO particles and 35 generations were used, and no segmentation noise was introduced in the rendered skin and edge maps. The obtained results (Figs. 3(e) and (f)) show that the performance improvement from one view to two views is significant. Adding more views improves the results noticeably but not significantly.

In order to assess the tolerance of the method to different levels of segmentation errors, all the rendered silhouette and edge maps were artificially corrupted with different levels of noise. The type of noise employed is similar to [7]. More specifically, positions are randomly selected within a map and the labels of all pixels in a circular neighborhood of a random radius are flipped. The aggregate measure of noise contamination is the percentage of pixels with swapped labels. In the plots of Figs. 3(g) and (h), the horizontal axis represents the percentage of noise-contaminated pixels in each skin map. Edge maps were contaminated with one third of this percentage. The contamination was applied independently to each artificial map r_s and r_e. In this experiment, 128 PSO particles and 35 PSO generations were used, and multiframes of eight views were considered.The plots indicate that the method exhibited robustness to moderate amounts of noise and failed for large amounts of noise. The exhibited robustness can be attributed to the large number of employed views. Since the noise of each view was assumed to be independent from all other views, the emerged consensus (over skin detection and edge detection) managed to cancel out low-variance noise. Figure 3 also demonstrates that the design choices regarding the objective function E (Sec. 2.3) are correct. This can be verified by the observed monotonic relation between E and the actual 3D hand pose estimation error.

Finally, Table 1 provides information on the runtime of these experiments. The table shows the number of multiframes per second for various parameterizations of the PSO (number of generations and number of particles per generation) and

Table 1. Number of multiframes per second processed for a number of PSO generations and camera views for 16/128 particles per generation

Generations	2 views	4 views	8 views
10	7.69/2.48	4.22/1.26	2.14/0.63
15	7.09/1.91	3.65/0.97	1.85/0.49
20	**6.23**/1.55	3.19/0.79	1.62/0.39
25	5.53/1.31	2.85/0.67	1.44/0.33
30	5.00/1.13	2.59/0.57	1.30/0.29
35	4.55/1.00	2.34/0.50	1.18/0.25
40	4.18/0.89	2.15/0.45	1.09/0.23

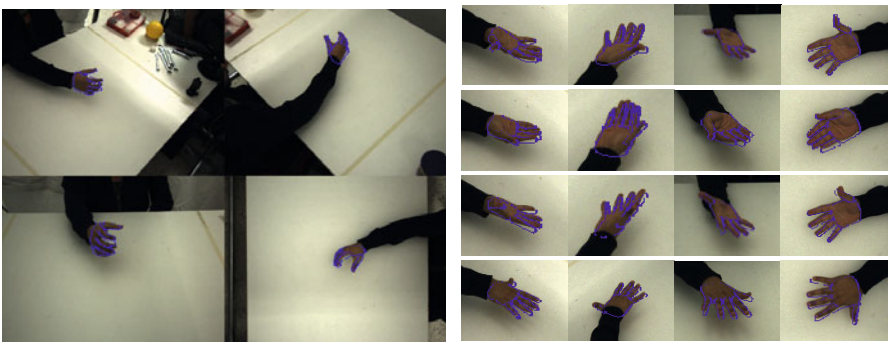


Fig. 4. Sample frames from real-world experiments. Left: four views of a multiframe of a cylindrical grasp. Right: Zoom on hands; Rows are from the same multiframe and columns correspond to the same camera view.

various number of views. The entry in boldface corresponds to 20 generations, 16 particles per generation and 2 views. According to the quantitative results presented earlier, this setup corresponds to the best trade-off between accuracy of results, computational performance and system complexity. This figure shows that the proposed method is capable of accurately and efficiently recovering the 3D pose of a hand observed from a stereo camera configuration at 6.2Hz. If 8 cameras are employed, the method delivers poses at a rate of 1.6Hz.

3.2 Experiments with Real World Images

Real-world image sequences were acquired using a multicamera system which is installed around a $2 \times 1m^2$ bench and consists of 8 *Flea2* PointGrey cameras. Cameras are synchronized by a timestamp-based software that utilizes a dedicated *FireWire 2* interface ($800\,MBits/sec$) which guarantees a maximum of $125\,\mu sec$ temporal discrepancy in images with the same timestamp. Each camera has a maximum framerate of 30 fps at highest (i.e. 1280×960) image resolution. The workstation where images are gathered has a quad-core Intel i7 920 CPU, 6 GBs RAM and an Nvidia GTX 295 dual GPU with $894\,GFlops$ processing power and 896 MBs memory per GPU core.

Several sequences of multiframes have been acquired, showing various types of hand activities such as isolated motions and hand-environment interactions including object grasping. Figure 4 provides indicative snapshots of 3D hand pose estimation superimposed on the original image data. Videos with results of these experiments are available online[1].

4 Discussion

In this paper, we proposed a novel method for the visual recovery of 3D hand pose of a human hand. This is formulated as an optimization problem which is accurately and robustly solved through Particle Swarm Optimization. In an effort to propose a method that is both accurate and computationally efficient, appropriate design choices were made to select components that exhibit data parallelism which is exploited by a GPU based implementation. The experimental evaluation in challenging datasets (complex hand articulation, distant hand views) demonstrates that accurate pose recovery can be achieved at a framerate that greatly outperforms the current state of the art. The individual constituents of the proposed method are clearly separated. It is quite easy for changes to be made to the objective function, the optimization method or the hand model without affecting the other parts. Current research if focused on considering more compact search spaces through the use of dimensionality reduction techniques.

Acknowledgements. This work was partially supported by the IST-FP7-IP-215821 project GRASP. The contributions of Asimina Kazakidi and Thomas Sarmis (members of the CVRL/ICS/FORTH) are gratefully acknowledged.

References

1. Moeslund, T.B., Hilton, A., Krüger, V.: A survey of advances in vision-based human motion capture and analysis. CVIU 104, 90–126 (2006)
2. Wang, R.Y., Popović, J.: Real-time hand-tracking with a color glove. ACM Transactions on Graphics 28, 1 (2009)
3. Erol, A., Bebis, G., Nicolescu, M., Boyle, R.D., Twombly, X.: Vision-based hand pose estimation: A review. CVIU 108, 52–73 (2007)
4. Athitsos, V., Sclaroff, S.: Estimating 3d hand pose from a cluttered image. In: CVPR, vol. 2, p. 432 (2003)
5. Rosales, R., Athitsos, V., Sigal, L., Sclaroff, S.: 3d hand pose reconstruction using specialized mappings. In: ICCV, pp. 378–385 (2001)
6. Wu, Y., Huang, T.S.: View-independent recognition of hand postures. In: CVPR, pp. 88–94 (2000)
7. Romero, J., Kjellstrom, H., Kragic, D.: Monocular real-time 3D articulated hand pose estimation. In: IEEE-RAS Int'l Conf. on Humanoid Robots, pp. 87–92 (2009)
8. Rehg, J.M., Kanade, T.: Visual tracking of high dof articulated structures: An application to human hand tracking. In: Eklundh, J.-O. (ed.) ECCV 1994. LNCS, vol. 801, pp. 35–46. Springer, Heidelberg (1994)

[1] http://www.ics.forth.gr/∼argyros/research/3Dhandpose.htm

9. Stenger, B., Mendonca, P., Cipolla, R.: Model-based 3D tracking of an articulated hand. In: CVPR, pp. II–310–II–315 (2001)
10. Sudderth, E., Mandel, M., Freeman, W., Willsky, A.: Visual hand tracking using nonparametric belief propagation. In: CVPR Workshop, pp. 189–189 (2004)
11. de la Gorce, M., Paragios, N., Fleet, D.: Model-based hand tracking with texture, shading and self-occlusions. In: CVPR, pp. 1–8 (2008)
12. John, V., Trucco, E., Ivekovic, S.: Markerless human articulated tracking using hierarchical particle swarm optimisation. Image and Vision Computing 28, 1530–1547 (2010)
13. Canny, J.: A computational approach to edge detection. PAMI 8, 679–698 (1986)
14. Argyros, A., Lourakis, M.: Real-time tracking of multiple skin-colored objects with a possibly moving camera. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3023, pp. 368–379. Springer, Heidelberg (2004)
15. Albrecht, I., Haber, J., Seidel, H.: Construction and animation of anatomically based human hand models. In: 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Eurographics Association, p. 109 (2003)
16. Turkowski, K.: Transformations of surface normal vectors. Technical report, Tech. Rep. 22, Apple Computer (July 1990)
17. Kennedy, J., Eberhart, R., Shi, Y.: Swarm intelligence. Morgan Kaufmann Publishers, San Francisco (2001)
18. Angeline, P.: Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences. In: Porto, V.W., Waagen, D. (eds.) EP 1998. LNCS, vol. 1447, pp. 601–610. Springer, Heidelberg (1998)
19. White, B., Shaw, M.: Automatically tuning background subtraction parameters using particle swarm optimization. In: IEEE ICME, pp. 1826–1829 (2007)
20. Clerc, M., Kennedy, J.: The particle swarm - explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation 6, 58–73 (2002)
21. Shaheen, M., Gall, J., Strzodka, R., Gool, L.V., Seidel, H.P.: A comparison of 3d model-based tracking approaches for human motion capture in uncontrolled environments. In: Workshop on Applications of Computer Vision, pp. 1–8 (2009)
22. Luo, Y., Duraiswami, R.: Canny edge detection on NVIDIA CUDA. In: CVPR 2008 Workshops, pp. 1–8 (2008)
23. Fischer, I., Gotsman, C.: Fast approximation of high-order Voronoi diagrams and distance transforms on the GPU. Journal of Graphics, GPU, & Game Tools 11, 39–60 (2006)
24. Pharr, M., Fernando, R.: Gpu gems 2: programming techniques for high-performance graphics and general-purpose computation (2005)

Stick It! Articulated Tracking Using Spatial Rigid Object Priors

Søren Hauberg and Kim Steenstrup Pedersen

The eScience Centre, Dept. of Computer Science, University of Copenhagen
{hauberg,kimstp}@diku.dk

Abstract. Articulated tracking of humans is a well-studied field, but most work has treated the humans as being independent of the environment. Recently, Kjellström et al. [1] showed how knowledge of interaction with a known rigid object provides constraints that lower the degrees of freedom in the model. While the phrased problem is interesting, the resulting algorithm is computationally too demanding to be of practical use. We present a simple and elegant model for describing this problem. The resulting algorithm is computationally much more efficient, while it at the same time produces superior results.

1 Introduction

Three dimensional articulated human motion tracking is the process of estimating the configuration of body parts over time from sensor input [2]. A large body of work have gone into solving this problem by using computer vision techniques without resorting to visual markers. The bulk of this work, however, completely ignores that almost all human movement somehow involves interaction with a rigid environment (people sit on *chairs*, walk on the *ground*, lift the *bottle* and so forth). By incorporating this fact of life, one can take advantage of the constraints provided by the environment, which effectively makes the problem easier to solve.

Recently, Kjellström et al. [1] showed that taking advantage of these constraints allows for improved tracking quality. To incorporate the constraints Kjellström et al., however, had to resort to a highly inefficient rejection sampling scheme. In this paper, we present a detailed analysis of this work and show how the problem can be solved in an elegant and computationally efficient manner. First we will, however, review the general articulated tracking framework and related work.

1.1 Articulated Tracking

Estimating the pose of a person using a single view point or a small baseline stereo camera is an inherently difficult problem due to self-occlusions. This manifests itself in that the distribution of the human pose is multi-modal with an unknown number of modes. Currently, the best method for coping with such

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 758–769, 2011.

distributions is the particle filter [3]. This aims at estimating the state of the system, which is represented as a set of weighted samples. These samples are propagated in time using a predictive model and assigned a weight according to a data likelihood. As such, the particle filter requires two subsystems: one for computing likelihoods by comparing the image data to a sample from the hidden state distribution, and one for predicting future states. In practice, the predictive system is essential in making the particle filter computationally feasible, as it can drastically reduce the number of needed samples. As an example, we shall later see how the predictive system can be phrased to incorporate constraints from the environment.

For the particle filter to work, we need a representation of the system state, which in our case is the human pose. As is common [2], we shall use the kinematic skeleton (see Fig. 1). This representation is a collection of connected rigid bones organised in a tree structure. Each bone can be rotated at the point of connection between the bone and its parent. We model the bones as having known constant length (i.e. rigid), so the direction of each bone constitute the only degrees of freedom in the kinematic skeleton. The direction in each joint can be parametrised with a vector of angles, noticing that different joints may have different number of degrees of freedom. We may collect all joint angle vectors into one large vector $\boldsymbol{\theta}_t$ representing all joint angles in the model at time t. The objective of the particle filter, thus, becomes to estimate $\boldsymbol{\theta}_t$ in each frame.

To represent the fact that bones cannot move freely (e.g. the elbow joint can only bend between 0 and 120 degrees), we restrict $\boldsymbol{\theta}_t$ to a subset $\boldsymbol{\Theta}$ of $\mathbb{R}^N$. In practice, $\boldsymbol{\Theta}$ is chosen such that each joint angle is restricted to an interval. This is often called box constraints [4].

From known bone lengths and a joint angle vector $\boldsymbol{\theta}_t$, it is straight-forward to compute the spatial coordinates of the bones. The root of the kinematic tree is placed at the origin of the coordinate system. The end point of the next bone along a branch in the tree is then computed by rotating the coordinate system and translating the root along a fixed axis relative to the parent bone. The rotation is parametrised by the angles of the joint in question and the length of the translation corresponds to the known length of the bone. We can repeat this process recursively until the entire kinematic tree has been traversed. This process is known as Forward Kinematics [5].

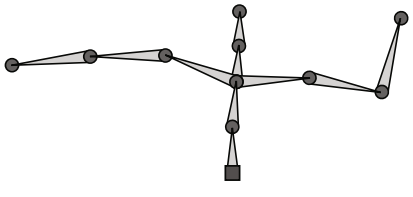


Fig. 1. An illustration of the kinematic skeleton. Circles correspond to the spatial bone end points and the square corresponds to the root.

1.2 Related Work

Most work in the articulated tracking literature falls in two categories. Either the focus is on improving the vision system or on improving the predictive system. Due to space constraints, we forgo a review of various vision systems as this paper is focused on prediction. For an overview of vision systems, see the review paper by Poppe [2].

Most work on improving the predictive system, is focused on learning motion specific priors, such as for *walking* [6,7,8,9,10,11,12]. Currently, the most popular approach is to restrict the tracker to some subspace of the joint angle space [10,8,9,13,7]. Such priors are, however, action specific. When no action specific knowledge is available it is common [14,1,10,15] to simply let $\boldsymbol{\theta}_t$ follow a normal distribution with a diagonal covariance, i.e.

$$p_{\text{gp}}(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}) \propto \mathcal{N}(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}, \text{diag}) \ \mathcal{U}_{\boldsymbol{\Theta}}(\boldsymbol{\theta}_t) \ , \tag{1}$$

where $\mathcal{U}_{\boldsymbol{\Theta}}$ is a uniform distribution on the legal set of angles that encodes the joint constraints. Recently, Hauberg et al. [16] showed that this model causes the spatial variance of the bone end points to increase as the kinematic chains are traversed. In practice this means that with this model the spatial variance of e.g. the hands is always larger than of the shoulders. We will briefly review a solution to this problem suggested by Hauberg et al. in Sec. 1.3, as it provides us a convenient framework for modelling interaction with the environment.

In general, as above, the environment is usually not incorporated in the tracking models. One notable environmental exception seems to be the ground plane [17,6]. Yamamoto and Yagishita [17] use a linear approximation of the motion path by linearising the forward kinematics function. As this is a highly non-linear function and motion paths in general are non-linear this modelling decision seems to be made out of sheer practicality. Promising results are, however, shown on constrained situations, such as when the position and orientation of a persons feet is known. Brubaker et al. [6] explicitly model the ground plane in a biomechanical model of walking. Their approach is, however, limited to interaction with the ground while walking.

Of particular importance to our work, is the paper by Kjellström et al. [1]. We will therefore review this in detail in Sec. 2.

1.3 Projected Spatial Priors

Recently, an issue with the standard general purpose prior from Eq. 1 was pointed out by Hauberg et al. [16]. Due to the tree structure of the kinematic skeleton, the spatial variance of bone end point increase as the kinematic chains are traversed. To avoid this somewhat arbitrary behaviour it was suggested to build the prior distribution directly in the spatial domain.

To define a predictive distribution in the spatial domain, Hauberg et al. first define a representation manifold $\mathcal{M} \in \mathbb{R}^{3L}$, where L denotes the number of bones. A point on this manifold corresponds to all spatial bone end points of a pose parametrised by a set of joint angles. More stringent, $\mathcal{M}$ can be defined as

$$\mathcal{M} = \{F(\boldsymbol{\theta}) | \boldsymbol{\theta} \in \boldsymbol{\Theta}\} \ , \tag{2}$$

where F denotes the forward kinematics function for the entire skeleton.

Once this manifold is defined, a Gaussian-like distribution can be defined simply by projecting a Gaussian distribution in $\mathbb{R}^{3L}$ onto $\mathcal{M}$, i.e.

$$p_{proj}(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{t-1}) = proj_{\mathcal{M}} \left[\mathcal{N}(F(\boldsymbol{\theta}_t) | F(\boldsymbol{\theta}_{t-1}), \Sigma)\right] \ . \tag{3}$$

When using a particle filter for tracking, one only needs to be able to draw samples from the prior model. This can easily be done by sampling from the normal distribution in $\mathbb{R}^{3L}$ and projecting the result onto $\mathcal{M}$. This, however, requires an algorithm for performing the projection. This is done by seeking

$$\hat{\boldsymbol{\theta}}_t = \min_{\boldsymbol{\theta}_t} \left|\left|\boldsymbol{x}_t - F(\boldsymbol{\theta}_t)\right|\right|^2 \qquad \text{s.t.} \qquad \boldsymbol{\theta}_t \in \boldsymbol{\Theta} \ , \tag{4}$$

where $\boldsymbol{x}_t$ denotes a sample from the normal distribution in $\mathbb{R}^{3L}$. This is an overdetermined constrained non-linear least-squares problem, that can be solved by any off-the-shelf optimisation algorithm [4]. We shall later see that the spatial nature of this prior is very helpful when designing priors that take the environment into account.

2 The KKB Tracker

Kjellström et al. [1] consider the situation where a person is holding on to a stick. It is assumed that the 3D position of the stick is known in each frame. In practice they track the stick using 8 calibrated cameras. They define the stick as

$$\text{stick}(\gamma_t) = \gamma_t \boldsymbol{a} + (1 - \gamma_t)\boldsymbol{b}, \quad \gamma_t \in [0, 1] \ , \tag{5}$$

where $\boldsymbol{a}$ and $\boldsymbol{b}$ are the end points of the stick.

The state is extended with a γ_t for each hand, which encodes the position of the respective hand on the stick. The state, thus, contains $\boldsymbol{\theta}_t$, $\gamma_t^{(\text{left})}$ and $\gamma_t^{(\text{right})}$. The goal is then to find an algorithm where the hand positions implied by $\boldsymbol{\theta}_t$ corresponds to the hand positions expressed by the γ_t's.

Kjellström et al. take a rejection sampling approach for solving this problem. They sample $\boldsymbol{\theta}_t$ from Eq. 1 and compute the attained hand positions using forward kinematics. They then keep generating new samples until the attained hand positions are within a given distance of the hand positions encoded by the γ_t's. Specifically, they keep generating new $\boldsymbol{\theta}_t$'s until

$$\|F_{\text{left}}(\boldsymbol{\theta}_t) - \text{stick}(\gamma_t^{(left)})\| < T_E \quad \text{and} \quad \|F_{\text{right}}(\boldsymbol{\theta}_t) - \text{stick}(\gamma_t^{(right)})\| < T_E \ , \tag{6}$$

where F_{left} is the forward kinematics function that computes the position of the left hand, F_{right} is the equivalent for the right hand and T_E is a threshold. We will denote this prior p_{kkb}, after the last names of its creators.

The γ_t's are also propagated in time to allow for sliding the hands along the stick. Specifically, Kjellström et al. let

$$p\left(\gamma_t^{(\text{left})}|\gamma_{t-1}^{(\text{left})}\right) \propto \mathcal{N}\left(\gamma_t^{(\text{left})}|\gamma_{t-1}^{(\text{left})}, \sigma^2\right) \; \mathcal{U}_{[0,1]}\left(\gamma_t^{(\text{left})}\right) \quad , \tag{7}$$

where $\mathcal{U}_{[0,1]}$ is the uniform distribution on $[0, 1]$. $\gamma_t^{(\text{right})}$ is treated the same way.

The advantage of this approach is that it actually works; successful tracking was reported in [1] and in our experience decent results can be attained with relatively few particles. Due to the rejection sampling, the approach is, however, computationally very demanding (see Sec. 5, in particular Fig. 4). The approach also has a limit on how many constraints can be encoded in the prior, as more constraints yield smaller acceptance regions. Thus, the stronger the constraints, the longer the running time. Furthermore, the rejection sampling has the side effect that the time it takes to predict one sample is not constant. In parallel implementations of the particle filter, such behaviour causes thread divergence, which drastically lessens the gain of using a parallel implementation.

3 Spatial Object Interaction Prior

We consider the same basic problem as Kjellström et al. [1], that is, assume we know the position of a stick in 3D and assume we know the person is holding on to the stick. As Kjellström et al., we extend the state with a γ_t for each hand that encodes where on the stick the hands are positioned using the model stated in Eq. 5. As before these are propagated in time using Eq. 7.

Following the idea of Hauberg et al. [16], we then define a motion prior in the spatial domain. Intuitively, we let each bone end point, except the hands, follow a normal distribution with the current bone end point as the mean value. The hands are, however, set to follow a normal distribution with a mean value corresponding to the hand position implied by $\gamma_t^{(\text{left})}$ and $\gamma_t^{(\text{right})}$. The resulting distribution is then projected back on the manifold $\mathcal{M}$ of possible poses, such that the final motion prior is given by

$$p_{\text{stick3d}}(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}) = proj_{\mathcal{M}}\left[\mathcal{N}(F(\boldsymbol{\theta}_t)|\boldsymbol{\mu}, \Sigma)\right] \quad , \tag{8}$$

where $\boldsymbol{\mu}$ indicates the just mentioned mean value. Samples can then be drawn from this distribution as described in Sec. 1.3.

3.1 Two Dimensional Object Information

When we defined p_{stick3d} we assumed we knew the three dimensional position of the stick. In the experiments presented in Sec. 5, we are using an active motion capture system to attain this information. While this approach might be feasible in laboratory settings it will not work in the general single-viewpoint setup; in practice it is simply too hard to accurately track even a rigid object in 3D. It is, however, not that difficult to track a stick in 2D directly in the image.

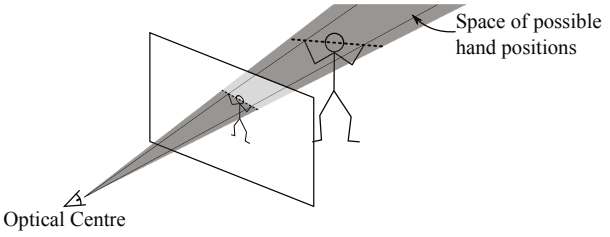


Fig. 2. An illustration of the geometry behind the p_{stick2d} model. The stick is detected in the image and the hands are restricted to the part of $\mathbb{R}^3$ that projects onto the detected stick.

We, thus, suggest a trivial extension of p_{stick3d} to the case were we only know the 2D image position of the stick.

From the 2D stick position in the image and the value of $\gamma_t^{(left)}$ we can compute the 2D image position of the left hand. We then know that the actual hand position in 3D must lie on the line going through the optical centre and the 2D image position. We then define the mean value of the predicted left hand as the projection of the current left hand 3D position onto the line of possible hand positions. The right hand is treated similarly. This is sketched in Fig. 2. The mean value of the remaining end point is set to their current position, and the resulting distribution is projected onto $\mathcal{M}$. We shall denote this motion prior p_{stick2d}.

4 Visual Measurements

To actually implement an articulated tracker, we need a system for making visual measurements. To keep the paper focused on prediction, we use a simple vision system [16] based on a consumer stereo camera[1]. This camera provides a dense set of three dimensional points $\mathbf{Z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_K\}$ in each frame. The objective of the vision system then becomes to measure how well a pose hypothesis matches the points. We assume that points are independent and that the distance between a point and the skin of the human follows a zero-mean Gaussian distribution, i.e.

$$p(\mathbf{Z}|\boldsymbol{\theta}_t) \propto \prod_{k=1}^{K} \exp\left(-\frac{\min\left[D^2(\boldsymbol{\theta}_t, \boldsymbol{z}_k),\ \tau\right]}{2\sigma^2}\right) , \tag{9}$$

where $D^2(\boldsymbol{\theta}_t, \boldsymbol{z}_k)$ denotes the squared distance between the point $\boldsymbol{z}_k$ and the skin of the pose $\boldsymbol{\theta}_t$ and τ is a constant threshold. The minimum operation is there to make the system robust with respect to outliers.

We also need to define the skin of a pose, such that we can compute distances between this and a data point. Here, we define the skin of a bone as a capsule with main axis corresponding to the bone itself. Since we only have a single view

[1] http://www.ptgrey.com/products/bumblebee2/

point, we discard the half of the capsule that is not visible. The skin of the entire pose is then defined as the union of these half-capsules. The distance between a point and this skin can then be computed as the smallest distance from the point to any of the half-capsules.

5 Experimental Results

Using the just mentioned likelihood model we can create an articulated tracker for each suggested prior. This gives us a set of weighted samples at each time step, which we reduce to one pose estimate $\hat{\boldsymbol{\theta}}_t$ by computing the weighted average.

Table 1. Results for the first sequence using 500 particles

Prior	Error (std.)	Computation Time
p_{kkb}	2.7 cm (1.3 cm)	687 sec./frame
p_{stick3d}	2.4 cm (1.0 cm)	108 sec./frame
p_{stick2d}	2.9 cm (1.5 cm)	108 sec./frame
p_{gp}	4.2 cm (2.3 cm)	96 sec./frame

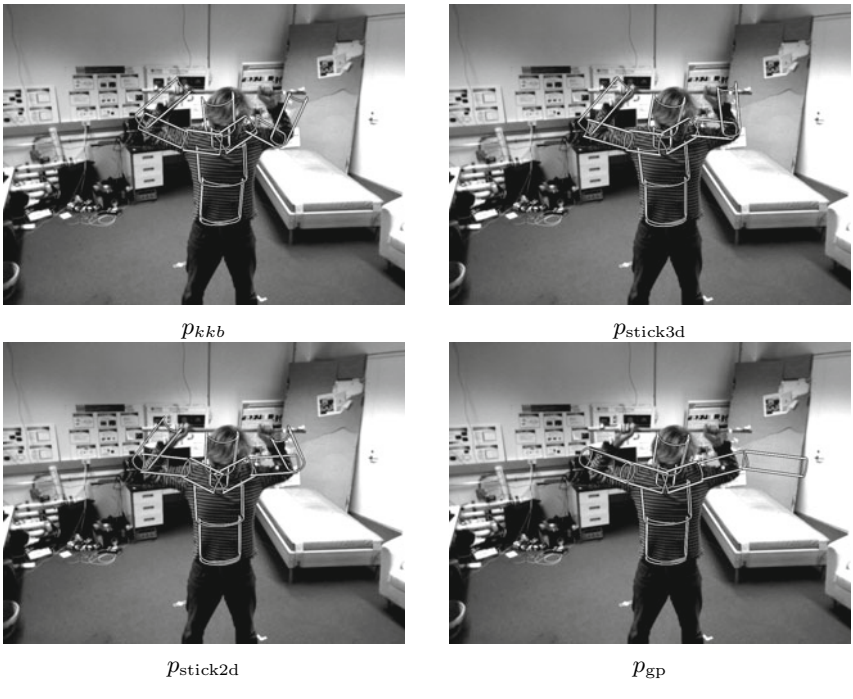
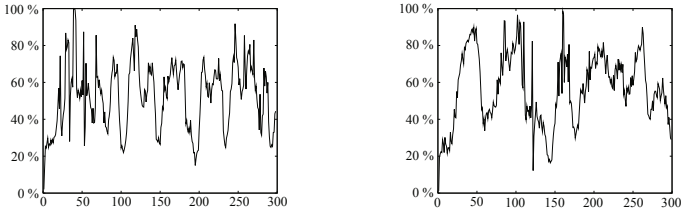


Fig. 3. Frame 182 from the first sequence. Image contrast has been enhanced for viewing purposes.



(a) First sequence (b) Second sequence

Fig. 4. Percentage of particles which reached the limit of the rejection sampling

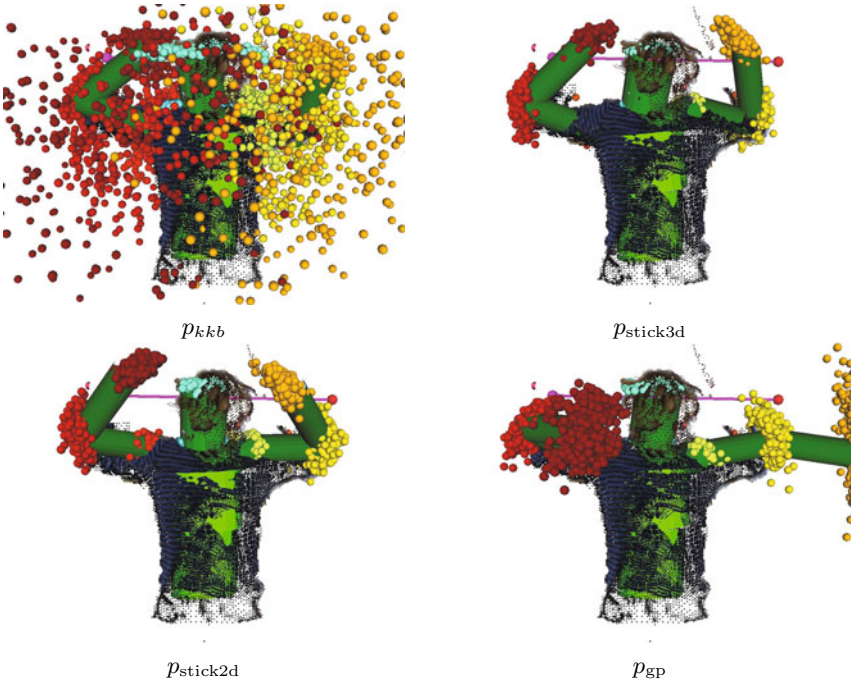


Fig. 5. The particles active in frame 182 in the first sequence

Table 2. Results for the second sequence using 500 particles

Prior	Error (std.)	Computation Time
p_{kkb}	8.4 cm (1.9 cm)	782 sec./frame
p_{stick3d}	2.2 cm (0.8 cm)	80 sec./frame
p_{stick2d}	2.8 cm (1.7 cm)	80 sec./frame
p_{gp}	8.4 cm (2.2 cm)	68 sec./frame

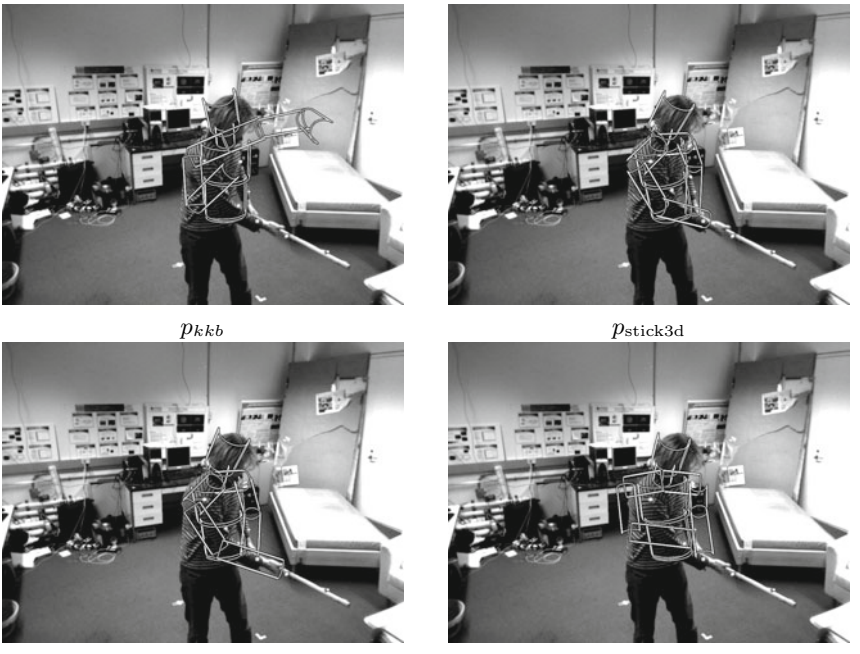


Fig. 6. Frame 101 from the second sequence. Image contrast has been enhanced for viewing purposes.

We record images from the previously mentioned stereo camera at 15 FPS along with synchronised data from an optical motion capture system[2]. We place motion capture markers on a stick such that we can attain its three dimensional position in each frame. In the case of $p_{stick2d}$, we only use the marker positions projected into the image plane.

To evaluate the quality of the attained results we also position motion capture markers on the arms of the test subject. We then measure the average distance between the motion capture markers and the capsule skin of the attained results. This measure is then averaged across frames, such that the error measure becomes

$$\mathcal{E} = \frac{1}{TM} \sum_{t=1}^{T} \sum_{m=1}^{M} D(\hat{\boldsymbol{\theta}}_t, \boldsymbol{v}_m) \ , \tag{10}$$

where $D(\hat{\boldsymbol{\theta}}_t, \boldsymbol{v}_m)$ is the Euclidean distance between the m^{th} motion capture marker and the skin at time t.

[2] http://www.phasespace.com/

In the first sequence we study a person who moves the stick from side to side and finally move the stick behind his head. This type of motion utilises the shoulder joints a lot, which is typically something that can cause difficulties for articulated trackers. We show selected frames from this sequence with the estimated pose superimposed in Fig. 3. Results are shown for the three different priors that utilise knowledge of the stick position. For reference, we also show the result of the standard model p_{gp} that assumes independent normally distributed joint angles. In all cases, 500 particles was used. As can be seen, the three stick-based priors all track the motion successfully, whereas the general purpose prior fail. This is more evident in the videos, which are available online[3].

To quantify the quality of the results, we compute the error measure from Eq. 10 for each of the attained results. This is reported along with the computation time in Table 1. As can be read, p_{stick3d} gives the most accurate results, closely followed by p_{kkb} and p_{stick2d}. However, when it comes to computation speed, we note that the p_{kkb} prior is 7.2 times slower than the general purpose angular prior, whereas our priors are both only 1.1 times slower.

Upon further study of the results attained by the p_{kkb} prior we note that in a few frames the pose estimate does not actually grab onto the stick. To understand this phenomena, we need to look at the details of the rejection sampling scheme. If we keep rejecting samples until Eq. 6 is satisfied, we have no way of guaranteeing that the algorithm will ever terminate. To avoid infinite loops, we stop the rejection sampling after a maximum of 5000 rejections. We found this to be a reasonable compromise between running times and accuracy. In Fig. 4a we plot the percentage of particles meeting the maximum number of rejections in each frame. As can be seen this number fluctuates and even reaches 100 percent in a few frames. This behaviour causes shaky pose estimates and even a few frames where the knowledge of the stick position is effectively not utilised. This can also be seen in Fig. 5 where the generated particles are shown for the different priors. Videos showing these are also available online[3]. Here we see that the p_{kkb} prior generates several particles with hand positions far away from the stick. We do not see such a behaviour of neither the $p_{stick3d}$ nor $p_{stick2d}$ priors.

We move on to the next studied sequence. Here the person is waiving the stick in a sword-fighting-manner. A few frames from the sequence with results superimposed are available in Fig. 6. While $p_{stick3d}$ and $p_{stick2d}$ are both able to successfully track the motion, p_{kkb} fails in several frames. As before, the reason for this behaviour can be found in the rejection sampling scheme. In Fig. 4b we show the percentage of particles reaching the maximum number of rejections. As before, we see that a large percentage of the particles often reach the limit and as such fail to take advantage of the known stick position. This is the reason for the erratic behaviour. In Table 2 we show accuracy and running time of the different methods, and here it is also clear that the p_{kkb} prior fails to track the motion even if it spends almost 10 times more time per frame than $p_{stick3d}$ and $p_{stick2d}$.

[3] http://humim.org/accv2010

6 Discussion

In this paper we have analysed an algorithm suggested by Kjellström et al. for articulated tracking when environmental constraints are available. We argued, and experimentally validated, that the algorithm is computationally too demanding to be of use in real-life settings. We then presented a simple model for solving the same problem, that only comes with a small computational overhead. The simplicity of our method comes from the decision to model the motion spatially rather than in terms of joint angles. This provides us with a general framework in which spatial knowledge can trivially be utilised. As most environmental knowledge is available in this domain, the idea can easily be extended to more complex situations.

In practice, much environmental information is not available in three dimensions, but can only be observed in the image plane. As such, we have suggested a straight-forward motion prior that only constraint limb positions in the image plane. This provides a framework that can actually be applied in real-life settings as it does not depend on three dimensional environmental knowledge that most often is only available in laboratory settings.

The two suggested priors are both quite simple and they encode the environmental knowledge in a straight-forward manner. The priors, thus, demonstrate the ease of which complicated problems can be solved when the motion is modelled spatially rather than in terms of joint angles. As spatial models have been shown to have more well-behaved variance structure than models expressed in terms of joint angles [16], we do believe spatial models can provide the basis of the next leaps forward for articulated tracking.

References

1. Kjellström, H., Kragić, D., Black, M.J.: Tracking people interacting with objects. In: CVPR 2010: Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2010)
2. Poppe, R.: Vision-based human motion analysis: An overview. Computer Vision and Image Understanding 108, 4–18 (2007)
3. Capp, O., Godsill, S.J., Moulines, E.: An overview of existing methods and recent advances in sequential Monte Carlo. Proceedings of the IEEE 95, 899–924 (2007)
4. Nocedal, J., Wright, S.J.: Numerical optimization. Springer Series in Operations Research. Springer, Heidelberg (1999)
5. Erleben, K., Sporring, J., Henriksen, K., Dohlmann, H.: Physics Based Animation. Charles River Media, Hingham (2005)
6. Brubaker, M.A., Fleet, D.J., Hertzmann, A.: Physics-based person tracking using the anthropomorphic walker. International Journal of Computer Vision 87, 140–155 (2010)
7. Wang, J.M., Fleet, D.J., Hertzmann, A.: Gaussian Process Dynamical Models for Human Motion. IEEE Transactions on Pattern Analysis and Machine Intelligence 30, 283–298 (2008)
8. Sminchisescu, C., Jepson, A.: Generative modeling for continuous non-linearly embedded visual inference. In: ICML 2004: Proceedings of the Twenty-First International Conference on Machine Learning, pp. 759–766. ACM, New York (2004)

9. Lu, Z., Carreira-Perpinan, M., Sminchisescu, C.: People Tracking with the Laplacian Eigenmaps Latent Variable Model. In: Platt, J., Koller, D., Singer, Y., Roweis, S. (eds.) Advances in Neural Information Processing Systems, vol. 20, pp. 1705–1712. MIT Press, Cambridge (2008)
10. Sidenbladh, H., Black, M.J., Fleet, D.J.: Stochastic tracking of 3D human figures using 2D image motion. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1843, pp. 702–718. Springer, Heidelberg (2000)
11. Elgammal, A.M., Lee, C.S.: Tracking People on a Torus. IEEE Transaction on Pattern Analysis and Machine Intelligence 31, 520–538 (2009)
12. Urtasun, R., Fleet, D.J., Fua, P.: 3D People Tracking with Gaussian Process Dynamical Models. In: CVPR 2006: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 238–245 (2006)
13. Urtasun, R., Fleet, D.J., Hertzmann, A., Fua, P.: Priors for people tracking from small training sets. In: Tenth IEEE International Conference on Computer Vision, vol. 1, pp. 403–410 (2005)
14. Bandouch, J., Engstler, F., Beetz, M.: Accurate human motion capture using an ergonomics-based anthropometric human model. In: Perales, F.J., Fisher, R.B. (eds.) AMDO 2008. LNCS, vol. 5098, pp. 248–258. Springer, Heidelberg (2008)
15. Balan, A.O., Sigal, L., Black, M.J.: A quantitative evaluation of video-based 3d person tracking. Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 349–356 (2005)
16. Hauberg, S., Sommer, S., Pedersen, K.S.: Gaussian-like spatial priors for articulated tracking. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6311, pp. 425–437. Springer, Heidelberg (2010)
17. Yamamoto, M., Yagishita, K.: Scene constraints-aided tracking of human body. In: CVPR, pp. 151–156. IEEE Computer Society, Los Alamitos (2000)

A Method for Text Localization and Recognition in Real-World Images

Lukas Neumann and Jiri Matas

Center for Machine Perception, Czech Technical University in Prague, Czech Republic

Abstract. A general method for text localization and recognition in real-world images is presented. The proposed method is novel, as it (i) departs from a strict feed-forward pipeline and replaces it by a hypotheses-verification framework simultaneously processing multiple text line hypotheses, (ii) uses synthetic fonts to train the algorithm eliminating the need for time-consuming acquisition and labeling of real-world training data and (iii) exploits Maximally Stable Extremal Regions (MSERs) which provides robustness to geometric and illumination conditions.

The performance of the method is evaluated on two standard datasets. On the Char74k dataset, a recognition rate of 72% is achieved, 18% higher than the state-of-the-art. The paper is first to report both text detection and *recognition* results on the standard and rather challenging ICDAR 2003 dataset. The text localization works for number of alphabets and the method is easily adapted to recognition of other scripts, e.g. cyrillics.

1 Introduction

Text localization and recognition in images of real-world scenes has received significant attention in the last decade [1, 2, 3, 4]. In contrast to text recognition in documents, which is satisfactorily addressed by state-of-the-art OCR systems [5], scene text localization and recognition is still an open problem. Factors contributing to the complexity of the problem include: non-uniform background, the need for compensation of perspective effects (for documents, rotation or rotation and scaling is sufficient); real-world texts are often short snippets written in different fonts and languages; text alignment does not follow strict rules of printed documents; many words are proper names which prevents an effective use of a dictionary.

Most published methods for text localization and recognition [1, 6, 7, 8] are based on sequential pipeline processing consisting of three steps - text localization, text segmentation and processing by an OCR for printed documents. In such approaches, the overall success rate of the method is a product of success rates of each stage as there is no possibility to refine decisions made by previous stages.

Some authors have focused on subtasks of the scene text recognition problem, such as text localization [3, 9, 10, 11, 4], individual character recognition [12, 13] or reading text from segmented areas of images [14]. Whilst they achieved promising

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 770–783, 2011.

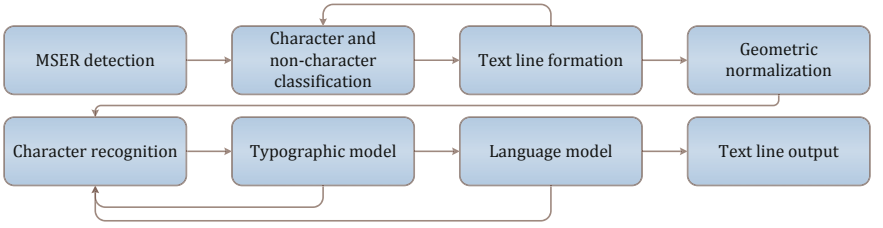


Fig. 1. Stages of the proposed method (incl. feedback loops for hypotheses verification)

results for individual subtasks, separating text localization from text recognition inevitably leads to loss of information, which results in degradation of overall text localization and recognition performance.

In this paper, we propose an end-to-end method for text localization and recognition. The technical contributions of the paper are the following. First, in the recognition part, no real-world training data are used. Learning is carried out directly on characters from fonts available in the Windows OS, with no preprocessing simulating acquisition effects, e.g. blur and deformations. Nevertheless, the proposed method achieves high recognition rates. Application of the method to other scripts, demonstrated on cyrillics in the paper, required only insertion of the relevant font sets (see Figure 2).

Second, characters are assumed to be extremal regions [15] in some scalar projection of pixel values. Character recognition is performed on a representation derived from the boundaries of extremal regions. Such a representation filters out effects of illumination, colour and texture variation in either foreground or background, or both, which is an important property for real-world text recognition (in contrast to printed document recognition, where such effects do not apply). Moreover, overlap of bounding boxes in tightly spaced text (e.g. with kerning) does not effect our method, which is not the case in methods where character detection is based on the sliding window. Extremal regions have been used for character recognition before [16], but in a very specific domain of single-font licence plate recognition rather than in a generic scene text recognition.

The proposed method is also novel in avoiding a pipeline architecture with a sequence of fixed decisions and working with multiple hypotheses at each stage

Fig. 2. Text localization and recognition output example on Russian text. Note: The only adjustment of the proposed method was a use of synthetic cyrillic fonts to train the character recognition with a Russian language model. The recognition is error free, with the exception of the exclamation mark which is not included in the training set.

of the processing (text localization, character segmentation, text line formation). Early steps are revisited in a hypothesis-verify framework and the decision about the most probable hypothesis is left to the last module, when values of all hidden parameters have been inferred.

The rest of the document is structured as follows: in Section 2, the problem of text detection and recognition is defined. Section 3 describes the proposed method. Performance evaluation of the proposed method is presented in Section 4. The paper is concluded in Section 5.

2 Problem Description

Let $\mathbf{I}$ be an input image and let $\mathcal{R}$ be a set of all contiguous regions of the image $\mathbf{I}$. Let S_m denote a set of all sequences of regions $S_m = \{(R_1, R_2, \ldots, R_m)\,; R_i \in \mathcal{R}\}$ of length m and let $\mathcal{S}$ denote a set of all sequences of all lengths $\mathcal{S} = \bigcup_{m=1\ldots n} S_m$, where n denotes the number of pixels in the image.

Text localization is defined as finding all sequences $s \in \mathcal{S}$ such that probability that the sequence represents a text $p_s(\text{text})$ has a local maximum, i.e. $\forall a \in \text{Adj}(s) : p_s(\text{text}) > p_a(\text{text})$ and $p_s(\text{text})$ is above a predefined threshold θ, where $\text{Adj}(s)$ denotes all sequences adjacent to sequence s. Two sequences are considered adjacent, if the first one differs from the second one by adding a single region at the end of the sequence. We assume that the probability $p_s(\text{text})$ is known from ground truth of training data.

Text recognition, given an alphabet $\mathcal{A}$, assigns a sequence of characters $\mathbf{y} = y_1 y_2 \ldots y_l : y_i \in \mathcal{A}$ to each sequence of regions s. Note that the length of the sequence of characters $\mathbf{y}$ may differ from the length of the sequence of regions s.

The problem of text localization and detection can be also described using notions of graph theory, which is more convenient for description of our method. Let $\mathbf{G}$ denote an undirected graph with vertices $V(\mathbf{G}) = \mathcal{R}$ and edges $E(\mathbf{G}) = \{(R_i, R_j) \in \mathcal{R} \times \mathcal{R} \mid i \neq j\}$. Each sequence $s \in \mathcal{S}$ of regions of length m is represented by a path $p = (v_1, v_2, \ldots v_n)\,; v_i \in V(\mathbf{G})$ in the graph $\mathbf{G}$ of the same length. The set of all sequences $\mathcal{S}$ then corresponds to the set of all paths $\mathcal{P}$ in the graph $\mathbf{G}$.

Because each path p has a one-to-one relation to a sequence s, the probability of a path p being a text equals to the probability of the corresponding sequence $p_s(\text{text})$. Let $h(p, v);\ p \in \mathcal{P}, v \in V(\mathbf{G})$ denote an auxiliary function such that

$$h(p, v) = \begin{cases} 1 & p_{pv}(\text{text}) > p_p(\text{text}) \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where pv denotes a path which was created by extending the path p with a vertex v.

Text localization can be then equally formulated as finding all paths $p \in \mathbf{P}$ such that $\forall v \in V(\mathbf{G}) : h(p, v) = 0$ and $|p| > l_{min}$, where l_{min} denotes a predefined threshold for minimal text length. In other words, text localization is a search for all paths in the graph $\mathbf{G}$ longer than l_{min}, such that extending the path by any other vertex decreases the probability of the path being a text.

3 Text Localization and Recognition

3.1 MSER Detection

Since the original search space induced by all regions $\mathcal{R}$ of image $\mathbf{I}$ is huge, certain approximations were applied in our approach. Assuming that individual characters are detected as Extremal Regions (ER) and taking computation complexity into consideration, the search space was limited to the set $\mathcal{M}$ of Maximally Stable Extremal Regions (MSER) [15], which can be computed in linear time in number of pixels [17].

The set of MSERs detected in certain scalar image projections (intensity, red channel, blue channel, green channel) defines the set of vertices of the graph $\mathbf{G}$, i.e. $V(\mathbf{G}) = \mathcal{M}$. The edges of the graph $\mathbf{G}$ are not stored explicitly, but they are induced on the fly (see Section 3.3).

3.2 Character and Non-character Classification

In this module, each vertex of graph $\mathbf{G}$ is labeled as a character or a non-character using a trained classifier which creates an initial hypothesis of text position, because character vertices are likely to be included in some path p representing a text.

The features used by the classifier (see Table 1) are scale invariant to detect all characters sizes, but they are not rotation invariant, which implies that characters at different rotations had to be included in the training set.

Once text lines are hypothesized (see Section 3.3), the initial character/non-character classifications are reassessed, taking hidden parameters of the text lines (character height, character spacing, etc.) into account. Thanks to the feed-back loop, the initial classification error has minimal impact on the overall performance.

A standard *Support Vector Machine* (SVM) [18] classifier with Radial Basis Function (RBF) kernel [19] was used. The classifier was trained on a set of 1227

(a) (b) (c) (d) (e)

Fig. 3. Character/non-character MSER classifier: (a) Character and (b) non-character training samples (MSER boundaries marked yellow). Initial MSER classification for (c) Armenian, (d) Russian and (e) Kannada script (character regions marked green, non-character regions marked red). Note: The training set contains only 1227 character samples of Latin script and 1396 non-character samples.

characters and 1396 non-characters obtained by manually annotating MSERs extracted from real-world images downloaded from Flickr. The classification error obtained by cross-validation was 5.6%. The training set is relatively small and certainly does not contain all possible fonts, scripts or even characters, but extending the training set with more examples did not bring any significant improvement in the classification success rate. This indicates that features used by the character classifier are insensitive to fonts and alphabets.

Table 1. Features used by the character classifier

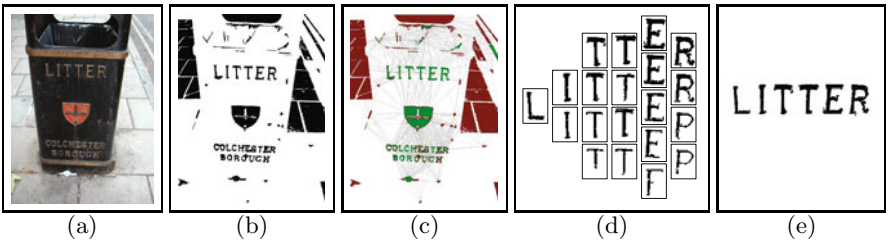
aspect ratio	relative segment height
compactness	number of holes
convex hull area to surface ratio	character color consistency
background color consistency	skeleton length to perimeter ratio

3.3 Text Line Hypothesis Formation

In real-world images a font rarely changes inside a word, which implies that certain character measurements (character height, aspect ratio, spacing between characters, stroke width, etc.) are either constant or constrained to a limited interval. Based on this observation, an approximation $\hat{h}(p, v)$ of function $h(p, v)$ (see Section 2) was implemented using a SVM classifier with polynomial kernel, whose feature vector is created by comparing average character measurements of the existing path p to the character measurements of given vertex v (see Table 2). The classifier was trained on the ICDAR 2003 Train set [20].

In our approach, only horizontal text areas which form a text line were considered. We think of a horizontal text line as a linear sequence of characters with straight or slightly curved bottom line, whose angle in the picture is in the range of ± 30 degrees.

Each path p is built in the following manner: The top-left unprocessed character vertex in the image is selected, creating an initial hypothesis of path p. The path p is then sequentially extended from left to right by all vertices $v \in V(\mathbf{G})$



(a) (b) (c) (d) (e)

Fig. 4. Text line hypothesis formation: (a) The source image. (b) MSERs detected in the red channel projection. (c) The induced graph (character vertices marked green, non-character marked red; edges longer than 300px omitted in the image for better readability) (d) Text line content hypotheses. (e) The selected hypothesis.

Table 2. Measurements used by the classifier in the approximation $\hat{h}(p, v)$

character width	character height
character surface	character color
aspect ratio	vertical distance from bottom line
stroke width	MSER margin [15]

such that $\hat{h}(p, v) = 1$ and distance of the vertex v in the source image is below the threshold d_{max}, which value was set experimentally to $3w_{max}$, where w_{max} denotes maximal character width in the existing path p.

If more than one vertex can be added to the path, multiple hypotheses about the path p are created and the decision about the most probable path is postponed for a later stage. If the path cannot be extended, all vertices of the path are marked as processed and next unprocessed top-left character vertex is selected to initialize a hypothesis of another independent path.

Every time a path p is extended by a new vertex, a bottom line approximation is calculated by Least-Median Squares (LMS) fitting of bottom points of individual regions in the text line; the approximation is then used to calculate the vertical distance of a vertex in the $\hat{h}(P, M)$ function. If the path is shorter than 5 vertices, only straight bottom line is allowed; if the path is longer, the bottom line is allowed to be slightly curved by fitting a parabola (see Figure 11, bottom-left).

3.4 Geometric Normalization

Perspective distortion is rectified prior to character recognition as all characters are trained in the frontoparallel view. The orientation of a camera to a plane with text in 3D space is modelled as a homography with a transformation matrix $\mathbf{H}$, which is decomposed as

$$\mathbf{H} = \begin{pmatrix} s\cos\theta & s\sin\theta & t_x \\ -s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1/b & -\sigma/b & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \ell_x & \ell_y & 1 \end{pmatrix} \tag{2}$$

The transformation has 8 degrees of freedom. However, only 3 of them are important for character recognition: the perspective foreshortening parameters ℓ_x, ℓ_y and the shear σ. Rotation θ can be easily calculated from the text line approximation and the scale parameters s and b, as well as the translation parameters t_x, t_y are not important thanks to the normalization, which is applied before the character recognition.

The sought parameters are estimated by the method of Myers et al. [21]. In this method, the perspective foreshortening parameters ℓ_x, ℓ_y are calculated from the horizontal vanishing point $\mathbf{V}_H$, which is located by finding top and bottom line of the text block and calculating its intersection.

Following Myers, the text block is rotated in the range of ± 3 degrees by 0.2 degree increments from detected text line orientation in order to find the top line.

For each rotation, the peak value of number of column top-most pixels in each row of the text block bitmap is calculated and the top points in the rotation with highest peak value are then considered a top line. The same process is repeated for the bottom line, here the number of column bottom-most pixels is calculated.

The shear σ is found by first rotating the text block so that the bottom line is horizontal and then iteratively applying a shear transformation in range of -45 to 45 degrees and measuring sum of squares of count of pixels in each column. The shear with the highest value is taken as a result.

(a) (b) (c)

Fig. 5. Geometrical normalization. (a) Text area in source image with detected top and bottom line. (b) Normalization input. (c) Normalization result.

3.5 Character Recognition

The character recognition starts by normalizing the MSER to a fixed-sized matrix of 35×35 pixel, while retaining the centroid of the region and aspect ratio [22]. Next, boundary pixels are inserted into separate bitmaps according to their orientation. After Gaussian blurring each bitmap is sub-sampled to a matrix of 5×5 pixels to generate 25 features. In total, 25 features $\times$ 8 directions generate 200 features for each MSER mask.

The 200-dimensional feature vectors are classified by a SVM classifier with Radial Basis Function (RBF) kernel. Based on the assumption that fonts in real-world images are very similar to standard synthetic fonts, the set of 40 synthetic fonts which are installed as part of Microsoft Windows OS was used to train the classifier using one-against-one strategy.

If the width of the region is bigger than threshold c_{min}, which was experimentally set to $c_{min} = 1.5w_{max}$, it is possible that the region actually corresponds to more than one character and thus an attempt to split the region is made. Candidate points for splitting are detected as the local minimum of the distance between the top and bottom pixels in a column (see Figure 8). Each combination of splitting is then evaluated in the context of surrounding letters using a feedback loop (see Section 3.7) and the hypothesis with the highest score according to the language model is selected.

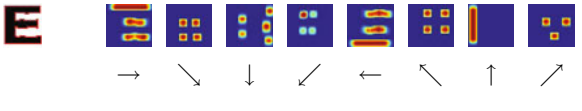
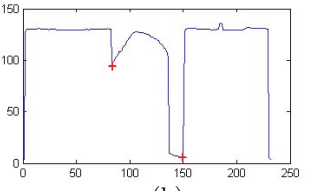


Fig. 6. Character recognition features: Input character (left). Features of the chain-code bitmap for each direction (right).

Fig. 7. Synthetic training samples of the character classifier. No "real world" training samples were used.

(a)



(b)

Split regions	Recognized text
	ItI
	Iu
	RI
	N

(c)

Fig. 8. Region splitting. (a) Source region. (b) Region column heights. (c) Resulting hypothesis.

3.6 The Typographic Model

A feed-back loop for character recognition was introduced as it is virtually impossible for the character classifier (see Section 3.5) to correctly differentiate between upper-case and lower-case variant of certain letters (such as "C" and "c") without knowing the heights of other letters in the text line. In order to correctly recognize the interchangeable letters, the height of unambiguously recognizable big and small letters is measured and then compared to actual height of the classified letter (see Figure 9).

Horizontal spacing between individual characters is measured and spaces between words are inserted at appropriate positions using a heuristics based on the analysis of the histogram of text line spacings.

Big	ABbDdEFfGHhKkLMNQRTtY012345678
Small	aegmnqry
Interchangeable	CcIiJjlOoPpSsUuVvWwXxZz

Fig. 9. The typographic model. Letter categories (left). Text line measurements (right) - (1) big and (2) small letters height, (3) base-line. Interchangeable letters marked gray.

3.7 The Language Model

The method treats each text line hypothesis individually, but in reality some of the hypotheses are mutually exclusive, either because their corresponding paths P in graph $\mathbf{G}$ have to be disjoint (one region can only be present in one text line) or due to their actual position in the image (a given area in an image can contain only one text line).

Given an alphabet $\mathcal{A}$, word $w = a_1 a_2 a_3 \ldots a_n, a_i \in \mathcal{A}$ and a set of words in a dictionary $\mathcal{W}$ a word score $s(w)$ is defined

$$s(w) = \begin{cases} 1 & w \in \mathcal{W} \\ \sqrt[n]{\prod_{i=1}^{n-1} P(a_i, a_{i+1})} & w \notin \mathcal{W} \end{cases} \tag{3}$$

The probability $P(r, s)$ is estimated using relative frequency of the sequence in the dictionary $\mathcal{W}$.

Given a text line $t = w_1, w_2, \ldots w_n$, the text line score $S(t)$ is then defined

$$S(t) = \sqrt[n]{\prod_{i=1}^{n} s(w_i)} \tag{4}$$

Given a set $\mathcal{T}$ of mutually exclusive hypotheses, the hypothesis with the highest score $S(t);\ t \in \mathcal{T}$ is selected.

Table 3. The set of mutually exclusive hypotheses and their score $S(t)$ in the English language model. The selected hypothesis is in bold.

Text line hypothesis	Recognized text	Score
LITTEP	LITTEP	0.0528
LITTER	LITTFR	0.0356
LITTER	LITTER	**0.0814**
LITTFP	LITTFP	0.0168

4 Experiments

4.1 Chars74K Dataset

The performance of the proposed method was evaluated on the *Chars74K*[1] dataset using the protocol proposed in the method of de Campos et. al [12]. In total, the *GoodImg* dataset used by the method of de Campos et. al contains 636 images with 7705 annotated characters of Latin alphabet. The SVM classifiers used in the method and their parameters were trained on an independent training set. The language model was created using a dictionary of approx. 10000 most frequent English words.

For each character in the ground truth, one of the three situations can occur: a letter is localized and recognized correctly (*matched*), a letter is localized correctly but not recognized correctly (*mismatched*) or a letter is not localized at all (*not found*). Since the dataset does not contain full annotations for words, it is not possible to obtain word recognition statistics.

[1] http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/

Fig. 10. Text localization and recognition examples on the Chars74K dataset. Kannada letters output marked red.

Table 4. Individual character recognition results on Chars74K dataset

	matched	mismatched	not found
proposed method	**71.6%**	**12.1%**	**16.3%**
de Campos et al.	54.3%	45.7%	N/A

The results show that the proposed method outperforms the results of de Campos et al. [12], where the best result achieved on the English *GoodImg* dataset is 54.30% correctly recognized letters. Note that the method of de Campos et al. works with manually located letters and thus there is no need for text localization. In our method, characters are detected automatically and the failure of detection is 16.3%, more than half of the total error rate of 28.4% (see Table 4).

Kannada letters in the Chars74k dataset were also successfully localized, but since the character classifier was not trained to support Kannada alphabet, the method outputs random strings for such texts (see Figure 10); since the method was evaluated only on English ground truth, the detected Kannada letters did not have any impact on the results.

4.2 ICDAR 2003 Dataset

The proposed method was also evaluated on ICDAR 2003 Robust Reading Competition Test dataset[2], which contains 5370 letters and 1106 words in 249 pictures. The same parameter setting as in the previous experiment (see Section 4.1) was used. The dictionaries supplied with the ICDAR 2003 dataset were not

[2] http://algoval.essex.ac.uk/icdar/Datasets.html

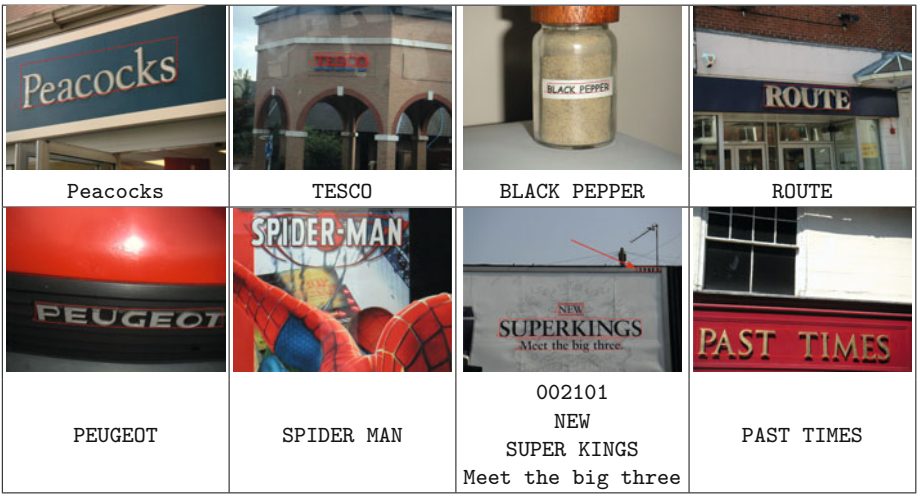


Fig. 11. Text localization and recognition examples on the ICDAR 2003 dataset

(a) (b) (c)

Fig. 12. Problems of the ICDAR 2003 ground truth. (a-b) Text detected by the proposed method but missed by the annotator (marked with a red arrow). (c) Interpretation as text is controversial (the cross is marked as "X" in the ground truth).

Fig. 13. Examples of ICDAR 2003 images where the proposed method fails to localize the text

Table 5. Results on the ICDAR 2003 dataset

(a) Text localization

method	precision	recall	f
Pen et. al [11]	0.67	0.71	0.69
Epshtein et. al [4]	0.73	0.60	0.66
Hinnerk Becker [23]	0.62	0.67	0.62
Alex Chen [23]	0.60	0.60	0.58
proposed method	**0.59**	**0.55**	**0.57**
Ashida [20]	0.55	0.46	0.50
HWDavid [20]	0.44	0.46	0.45
Wolf [20]	0.30	0.44	0.35
Qiang Zhu [23]	0.33	0.40	0.33
Jisoo Kim [23]	0.22	0.28	0.22
Nobuo Ezaki [23]	0.18	0.36	0.22
Todoran [20]	0.19	0.18	0.18

(b) Robust reading

method	precision	recall	f	t
proposed method	0.42	0.39	0.40	89s

(c) Individual character recognition (total numbers in parentheses)

	matched	mismatched	not found
proposed method	67.0% (3598)	12.9% (695)	20.1% (1077)

used in order to evaluate generic performance of the method. The standard definitions of word precision and recall defined in ICDAR 2003 Text Locating and Robust Reading competitions were used [20].

The results show that in terms of text localization, the proposed method achieves worse results than the winner algorithm of ICDAR 2005 [23] or the method proposed by Pen et al. [11], but is still competitive. In text recognition evaluation, we are not able to compare the proposed method with any existing method because there were no entries for ICDAR 2003/2005 Robust Reading competitions. We are not aware of any method with results on the complete ICDAR 2003 dataset.

5 Conclusions

An end-to-end method for scene text localization and recognition was proposed. The proposed method introduces a number of novel features, mainly: a departure from a strict feed-forward pipeline that is replaced by a hypotheses-verification framework simultaneously processing multiple text line hypotheses; the use of synthetic fonts to train the algorithm eliminating the need for time-consuming acquisition and labeling of real-world training data and the use of MSERs which provides robustness to geometric and illumination conditions.

The performance of the method was evaluated on two standard datasets. On the de Campos et al. Char74k dataset [12], a highly significant increase in recognition rate from 53% [12] to 72% was achieved. The text recognition results on the ICDAR 2003 dataset ($f = 0.40$, 67.0% correctly recognized letters) establishes a new baseline as no results in Robust Reading on a complete ICDAR 2003 dataset have been published.

The text localization results on the ICDAR 2003 dataset ($f = 0.57$) are worse than the method proposed by Pen et al. [11] ($f = 0.69$). Most frequent

problems of the proposed method in text localization are individual letters not being detected as MSERs in the projections used, invalid text line formation or invalid word breaking. However, the result has to be interpreted carefully as we noticed that there are problems with the ICDAR 2003 evaluation protocol, e.g. not all text in the image is marked as such and vice versa (see Figure 12).

Acknowledgement. The authors were supported by EC project FP7-ICT-247022 MASH, by Czech Government reseach program MSM6840770038 and by Grant Agency of the CTU Prague project SGS10/069/OHK3/1T/13.

References

1. Wu, V., Manmatha, R., Riseman Sr., E.M.: Textfinder: An automatic system to detect and recognize text in images. IEEE Trans. Pattern Anal. Mach. Intell. (1999)
2. Chen, X., Yang, J., Zhang, J., Waibel, A.: Automatic Detection and Recognition of Signs From Natural Scenes. IEEE Trans. on Image Processing 13, 87–99 (2004)
3. Ezaki, N.: Text detection from natural scene images: towards a system for visually impaired persons. In: Int. Conf. on Pattern Recognition, pp. 683–686 (2004)
4. Epshtein, B., Ofek, E., Wexler, Y.: Detecting text in natural scenes with stroke width transform. In: CVPR 2010: Proc. of the 2010 Conference on Computer Vision and Pattern Recognition (2010)
5. Lin, X.: Reliable OCR solution for digital content re-mastering. In: Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series (2001)
6. Chen, X., Yuille, A.L.: Detecting and reading text in natural scenes. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 366–373 (2004)
7. Gao, J., Yang, J.: An adaptive algorithm for text detection from natural scenes. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, p. 84 (2001)
8. Jain, A.K., Yu, B.: Automatic text location in images and video frames. In: International Conference on Pattern Recognition, vol. 2, p. 1497 (1998)
9. Pan, Y.F., Hou, X., Liu, C.L.: A robust system to detect and localize texts in natural scene images. In: IAPR International Workshop on Document Analysis Systems, pp. 35–42 (2008)
10. Kim, E., Lee, S., Kim, J.: Scene text extraction using focus of mobile camera. In: International Conference on Document Analysis and Recognition, pp. 166–170 (2009)
11. Pan, Y.F., Hou, X., Liu, C.L.: Text localization in natural scene images based on conditional random field. In: ICDAR 2009: Proc. of the 2009 10th International Conference on Document Analysis and Recognition, pp. 6–10 (2009)
12. de Campos, T.E., Babu, B.R., Varma, M.: Character recognition in natural images. In: VISAPP, February 05-08 (2009)
13. Yokobayashi, M., Wakahara, T.: Segmentation and recognition of characters in scene images using selective binarization in color space and gat correlation. In: Proc. of the 8th International Conference on Document Analysis and Recognition, pp. 167–171 (2005)
14. Weinman, J.J., Learned-Miller, E., Hanson, A.R.: Scene text recognition using similarity and a lexicon with sparse belief propagation. IEEE Trans. Pattern Anal. Mach. Intell. 31, 1733–1746 (2009)

15. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. Image and Vision Computing 22, 761–767 (2004)
16. Matas, J(G.), Zimmermann, K.: A new class of learnable detectors for categorisation. In: Kalviainen, H., Parkkinen, J., Kaarna, A. (eds.) SCIA 2005. LNCS, vol. 3540, pp. 541–550. Springer, Heidelberg (2005)
17. Nistér, D., Stewénius, H.: Linear time maximally stable extremal regions. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 183–196. Springer, Heidelberg (2008)
18. Cristianini, N., Shawe-Taylor, J.: An introduction to Support Vector Machines. Cambridge University Press, Cambridge (2000)
19. Muller, K.R., Mika, S., Ratsch, G., Tsuda, K., Scholkopf, B.: An introduction to kernel-based learning algorithms. IEEE Trans. on Neural Networks 12, 181–201 (2001)
20. Lucas, S.M., Panaretos, A., Sosa, L., Tang, A., Wong, S., Young, R.: Icdar 2003 robust reading competitions. In: ICDAR 2003: Proc. of the 7th International Conference on Document Analysis and Recognition, p. 682 (2003)
21. Myers, G.K., Bolles, R.C., Luong, Q.T., Herson, J.A., Aradhye, H.: Rectification and recognition of text in 3-d scenes. IJDAR 7, 147–158 (2005)
22. Liu, C.L., Nakashima, K., Sako, H., Fujisawa, H.: Handwritten digit recognition: investigation of normalization and feature extraction techniques. Pattern Recognition 37, 265–279 (2004)
23. Lucas, S.M.: Text locating competition results. In: International Conference on Document Analysis and Recognition, pp. 80–85 (2005)

Author Index